

Jan Camenisch  
Simone Fischer-Hübner  
Kai Rannenberg  
*Editors*

# Privacy and Identity Management for Life



 Springer



# Privacy and Identity Management for Life



Jan Camenisch • Simone Fischer-Hübner  
Kai Rannenberg  
Editors

# Privacy and Identity Mangement for Life

 Springer



*Editors*

Jan Camenisch  
IBM Research - Zürich  
Säumerstrasse 4  
8803 Rüschlikon  
Switzerland  
jca@zurich.ibm.com

Simone Fischer-Hübner  
Universitetsgatan 2  
65188 Karlstad  
Sweden  
simone.fischer-huebner@kau.se

Kai Rannenberg  
Goethe University Frankfurt  
T-Mobile Chair of Mobile Business & Multilateral  
Security  
Grueneburgplatz 1  
60629 Frankfurt/Main  
Germany  
Kai.Rannenberg@m-chair.net  
www.m-chair.net

ISBN 978-3-642-20316-9 e-ISBN 978-3-642-20317-6  
DOI 10.1007/978-3-642-20317-6  
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011930188

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

*Cover design:* deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

*To the memory of our colleague and friend Andreas Pfitzmann*

The protection of people's privacy was dear to Andreas' heart. He was an inspiration to all of us, and many of the results presented in this book owe to him.





# Preface

*Publication is a self-invasion of privacy.*

Marshall McLuhan

Individuals in the Information Society want to safeguard their autonomy and retain control over their personal information, irrespective of their activities. Information technologies generally do not consider those user requirements, thereby putting the privacy of the citizen at risk. At the same time, the Internet is changing from a client-server to a collaborative paradigm. Individuals are contributing throughout their life leaving a life-long trace of personal data. This raises substantial new privacy challenges.

*Saving digital privacy.* By 2008, the European project PRIME (Privacy and Identity Management for Europe) had demonstrated that existing privacy technologies can enable citizens to execute their legal rights to control their personal information in on-line transactions. It had raised considerable awareness amongst stakeholders and has significantly advanced the state of the art in the areas of privacy and identity management. PrimeLife has been building on the momentum created and the results achieved by PRIME to address emerging challenges in the areas of privacy and identity management and really bring privacy and identity management to live:

- A first, short-term goal of PrimeLife was to provide scalable and configurable privacy and identity management in new and emerging internet services and applications such as virtual communities and Web 2.0 collaborative applications.
- A second, longer-term goal of PrimeLife was to protect the privacy of individuals over their whole span of life. Each individual leaves a multitude of traces during a lifetime of digital interactions. Technological advancements facilitate extensive data collection, unlimited storage, as well as reuse and life-long linkage of these digital traces.
- A third goal of PrimeLife was to support privacy and identity management by progressing the state of the art on
  - tools guaranteeing privacy and trust,



- the usability experience of privacy and identity management solutions,
  - security and privacy policy systems, and
  - privacy-enabling infrastructures.
- The last but certainly important goal of PrimeLife was to disseminate our results and enable their use in real life. We organized interdisciplinary Summer Schools of Privacy, organized and participated in standardization groups and meetings, and made the source code and documentation of most of our prototypes and implementations available for free use.

*This Book.* After more than three years of work in PrimeLife, this book aims at giving an overview of the results achieved. It is therefore structured into an introduction and six parts covering the most important areas of privacy and identity management considering the life of today: Several aspects of “Privacy in Life” are discussed in Part I, followed by Part II “Mechanisms for Privacy”. Part III is dedicated to “Human Computer Interaction (HCI)”, and Part IV to “Policy Languages”. Part V focuses on “Infrastructures for Privacy and Identity Management,” before Part VI “Privacy Live” comes full circle describing how PrimeLife is reaching out to the life of today’s and the future’s netizens.

*Acknowledgements.* Editing and writing a multidisciplinary book like this is not possible without many contributors that dedicated their knowledge, expertise, and time to make the work a success. Among the many people that worked on this PrimeLife book, we would like to give a special thanks to:

- The partner organisations and individual researchers in PrimeLife that contributed to this book and the underlying PrimeLife deliverables. They come from seven Member States of the European Union, Switzerland, and the USA.
- Els van Herreweghen who was instrumental in writing and submitting the project proposal and Dieter Sommer who did a wonderful job as coordinator.
- The international team of reviewers that as friends of PrimeLife dedicated their time to improve and streamline the parts of this book. We are particularly grateful to Bart De Decker, Ian Glazer, David-Olivier Jaquet-Chiffelle, Vaclav Matyas, Vincent Naessens, Lasse Øverlier, Jakob Pagter, Michael Pedersen, Claire Vishik, and Jozef Vyskoc for their most valuable comments and suggestions.
- The people who helped to create this book:
  - Michelle Cryer for transforming much of the nerds’ English in the draft versions of this book into English.
  - The PrimeLife “Activity Leaders” Katrin Borcea-Pfzmann, Sandra Steinbrecher, Pierangela Samarati, Gregory Neven, Gökhan Bal, and Marit Hansen for their support and coordination of the respective parts. Gregory also provided the Latex templates and wisdom on how to put the different chapters and parts together into this book.

- Our colleagues at IBM Research – Zurich, Karlstad University, and Goethe University Frankfurt for keeping our backs covered when we were working on this book.
- Saskia Werdmüller for the final checking of the manuscript.
- Christian Rauscher at Springer for helping us with publication of the book.
- The European Commission for funding PrimeLife and our Project Officers Manuel Carvalhosa and Maria-Concepcion Anton-Garcia, their management, Jacques Bus, Jesus Villasante, Thomas Skordas, Gustav Kalbe, and Dirk van Rooy, and the project reviewers Jean-Marc Dinant, Alberto Escudero-Pascual, and Massimo Felici for most valuable feedback and encouragement.
- Last, but certainly not least, all the members of the PrimeLife Reference Group for many inspiring discussions and lots of valuable and helpful feedback.

We hope that this book will contribute to the understanding that digital privacy can be a reality: Many tools exist and are waiting to be deployed in practice.

Zurich, Karlstad, Frankfurt,  
March 2011

*Jan Camenisch  
Simone Fischer-Hübner  
Kai Rannenberg*



## List of Contributors to this Book

**Stefano Paraboschi, Gerardo Pelosi, Mario Verdicchio**

Università degli Studi di Bergamo, DIIMM  
Viale Marconi 25, Dalmine, I-24044

**Cornelia Graf, Christina Hochleitner, Manfred Tscheligi, Peter Wolkerstorfer**

CURE – Center for Usability Research and Engineering  
Modecenterstrasse 17, Vienna, A-1110

**Katrin Borcea-Pfitzmann, Jaromir Dobias, Benjamin Kellermann, Stefan Köpsell, Andreas Pfitzmann, Stefanie Pötzsch, Sandra Steinbrecher**

Dresden University of Technology, Department of Computer Science  
Nöthnitzer Strasse 46, Dresden, D-01187

**Marc-Michael Bergfeld, Stephan Spitz**

Giesecke & Devrient  
Prinzregentenstrasse 159, München, D-81677

**Gökhan Bal, Sascha Koschinat, Kai Rannenberg, Christian Weber**

Goethe University Frankfurt, T-Mobile Chair of Mobile Business & Multilateral Security  
Grüneburgplatz 1, Frankfurt/Main, D-60629

**Jan Camenisch, Maria Dubovitskaya, Gregory Neven, Franz-Stefan Preiss**

IBM Research – Zurich  
Säumerstrasse 4, Rüschlikon, CH-8803

**Julio Angulo**

Karlstad University, Department of Information Systems  
Universitetsgatan 2, Karlstad, S-65188

**Simone Fischer-Hübner, Hans Hedbom, Tobias Pulls**

Karlstad University, Department of Computer Science  
Universitetsgatan 2, Karlstad, S-65188

**Erik Wästlund**

Karlstad University, Department of Psychology  
Universitetsgatan 2, Karlstad, S-65188

**Filipe Beato, Markulf Kohlweiss, Jorn Lapon, Stefan Schiffner, Karel Wouters**

Katholieke Universiteit Leuven, ESAT - COSIC  
Kasteelpark Arenberg 10, Leuven, B-3001

**Laurent Bussard, Ulrich Pinsdorf**

Microsoft EMIC  
Ritterstrasse 23, Aachen, D-52072

**Claudio A. Ardagna, Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, Eros Pedrini, Pierangela Samarati**

Università degli Studi di Milano, Dip. di Tecnologie dell'Informazione  
Via Bramante 65, Crema, I-26013

**Michele Bezzi, Akram Njeh, Stuart Short, Slim Trabelsi**

SAP Labs France

805, Avenue du Docteur Maurice Donat, BP 1216, Mougins Cedex, F-06254

**Bibi van den Berg, Laura Klaming, Ronald Leenes, Arnold Roosendaal**

Universiteit van Tilburg, TILT

Postbus 90153, Tilburg, NL-5000 LE

**Marit Hansen, Leif-Erik Holtz, Ulrich König, Sebastian Meissner, Jan Schallaböck, Katalin Storf, Harald Zwingelberg**

Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein  
Holstenstrasse 98, Kiel, D-24103

**Carine Bournez, Dave Raggett, Rigo Wenning**

World Wide Web Consortium (W3C)

Route des Lucioles, BP 93, Sophia-Antipolis, F-06902



# Contents

## Introduction

<b>1</b>	<b>PrimeLife</b> .....	5
	Andreas Pfitzmann, Katrin Borcea-Pfitzmann, and Jan Camenisch	
1.1	Motivation .....	5
1.2	Vision and Objectives of the PrimeLife Project .....	7
1.3	Defining Privacy .....	8
1.4	From Identity via Identity Management to Privacy by Identity Management .....	9
1.4.1	Identity – What it is .....	10
1.4.2	Presentation of Identities – Pseudonyms .....	13
1.4.3	Time Aspects of Identity Management and Privacy .....	17
1.5	Further Facets of Privacy .....	19
1.6	PrimeLife’s Contributions to Protect Privacy .....	20
1.6.1	Part I - Privacy in Life .....	22
1.6.2	Part II - Mechanisms for Privacy .....	22
1.6.3	Part III - Human Computer Interaction (HCI) .....	23
1.6.4	Part IV - Policy Languages .....	24
1.6.5	Part V- Infrastructures for Privacy and Identity Management ..	25
1.6.6	Part VI- Privacy Live .....	25
	<b>References Introduction</b> .....	27

## Part I Privacy in Life

<b>2</b>	<b>Privacy in Social Software</b> .....	33
	Bibi van den Berg, Stefanie Pötzsch, Ronald Leenes, Katrin Borcea-Pfitzmann, and Filipe Beato	
2.1	Scenarios and Requirements .....	33
2.1.1	Scenario 1: A Social Network Site .....	35
2.1.2	Scenario 2: A Forum .....	36

2.1.3	General Requirements . . . . .	36
2.2	Two Prototypes for Privacy-Enhanced Social Networking . . . . .	37
2.2.1	Introduction . . . . .	37
2.2.2	Privacy Issues in Social Network Sites . . . . .	38
2.2.3	Clique: An Overview . . . . .	42
2.2.4	Scramble!: An Overview . . . . .	46
2.3	Privacy-Enhancing Selective Access Control for Forums . . . . .	50
2.3.1	Objectives . . . . .	50
2.3.2	Introducing phpBB Forum Software and PRIME Framework . . . . .	51
2.3.3	Extending phpBB with Selective Access Control . . . . .	52
2.3.4	Scenario Revisited . . . . .	54
2.3.5	Privacy-Awareness Information . . . . .	55
2.3.6	User Survey . . . . .	55
2.4	Concluding Remarks . . . . .	59
2.5	Acknowledgements . . . . .	60
<b>3</b>	<b>Trustworthiness of Online Content . . . . .</b>	<b>61</b>
	Jan Camenisch, Sandra Steinbrecher, Ronald Leenes, Stefanie Pöttsch, Benjamin Kellermann, and Laura Klaming	
3.1	Introduction . . . . .	61
3.2	Scenarios and requirements . . . . .	63
3.2.1	Scenarios . . . . .	63
3.2.2	High-level mechanisms . . . . .	65
3.2.3	Requirements of mechanisms . . . . .	66
3.3	Experiments . . . . .	70
3.3.1	Binding metadata to data . . . . .	71
3.3.2	User Reputation and Certification . . . . .	74
3.4	Demonstrators . . . . .	76
3.4.1	Trustworthy Blogging . . . . .	76
3.4.2	Encouraging Comments with Incentives . . . . .	78
3.4.3	Author reputation system and trust evaluation of content in MediaWiki . . . . .	80
3.5	Conclusive Remarks . . . . .	84
3.6	Acknowledgements . . . . .	85
<b>4</b>	<b>Identity and Privacy Issues Throughout Life . . . . .</b>	<b>87</b>
	Jaromir Dobias, Marit Hansen, Stefan Köpsell, Maren Raguse, Arnold Roosendaal, Andreas Pfitzmann, Sandra Steinbrecher, Katalin Storf, and Harald Zwingelberg	
4.1	Challenges and Requirements . . . . .	87
4.1.1	Dealing with Dynamics . . . . .	87
4.1.2	Digital Footprint . . . . .	91
4.1.3	Concepts for Delegation . . . . .	94
4.2	Demonstrator . . . . .	99
4.2.1	Overview of the Backup Demonstrator Architecture . . . . .	102



4.2.2	Deployment and Usage of the Demonstrator . . . . .	109
4.3	Concluding Remarks . . . . .	110
4.4	Acknowledgements . . . . .	110
<b>References Part I . . . . .</b>		<b>111</b>
<b>Part II Mechanisms for Privacy</b>		
<b>5</b>	<b>Cryptographic Mechanisms for Privacy . . . . .</b>	<b>117</b>
Jan Camenisch, Maria Dubovitskaya, Markulf Kohlweiss, Jorn Lapon, and Gregory Neven		
5.1	Introduction . . . . .	117
5.2	Cryptography to the Aid . . . . .	118
5.3	Private Credentials, Their Extensions, and Applications . . . . .	119
5.3.1	Extended Functionalities . . . . .	120
5.3.2	Direct Anonymous Attestation . . . . .	123
5.4	Other Privacy-Enhancing Authentication Mechanisms . . . . .	123
5.4.1	Privacy-Enhancing Encryption . . . . .	126
5.5	Electronic Voting, Polling, and Petitions . . . . .	127
5.6	Oblivious Transfer with Access Control and Prices . . . . .	128
5.7	Oblivious Trusted Third Parties . . . . .	130
5.8	Conclusion . . . . .	134
<b>6</b>	<b>Transparency Tools . . . . .</b>	<b>135</b>
Hans Hedbom, Tobias Pulls, and Marit Hansen		
6.1	Introduction . . . . .	135
6.2	Setting the Scene . . . . .	137
6.3	On Privacy Preserving and Secure Logs . . . . .	138
6.3.1	Attacker Model and Security Evaluation . . . . .	139
6.4	Prior Work and Our Contribution . . . . .	139
6.5	Technical Overview . . . . .	140
6.5.1	State and Secrets . . . . .	140
6.5.2	Entry Structure and Storage . . . . .	141
6.5.3	API . . . . .	142
6.5.4	Unlinkability . . . . .	142
6.6	Conclusion and Outlook . . . . .	143
<b>7</b>	<b>Interoperability of Trust and Reputation Tools . . . . .</b>	<b>145</b>
Sandra Steinbrecher and Stefan Schiffner		
7.1	Introduction . . . . .	145
7.2	Social need . . . . .	146
7.3	Legal Aspect . . . . .	147
7.4	Security and Privacy Requirements . . . . .	148
7.5	Technical Implementability . . . . .	149
7.6	Infrastructure . . . . .	150
7.6.1	Interoperability with Applications . . . . .	150

7.6.2	Interoperability with Trust Management . . . . .	152
7.6.3	Interoperability with Identity Management . . . . .	153
7.6.4	Resulting implementation . . . . .	154
7.7	Conclusion . . . . .	155
<b>8</b>	<b>Data Privacy . . . . .</b>	<b>157</b>
	Michele Bezzi, Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, Stefano Paraboschi, and Pierangela Samarati	
8.1	Introduction . . . . .	157
8.2	Privacy Metrics and Information Theory . . . . .	158
8.2.1	Basic Concepts . . . . .	159
8.2.2	Traditional Privacy Metrics . . . . .	160
8.2.3	An Information Theoretic Approach for Privacy Metrics . . . . .	161
8.2.4	Protecting Privacy of Sensitive Value Distributions . . . . .	164
8.3	Privacy Protection Techniques . . . . .	165
8.3.1	Basic Concepts . . . . .	165
8.4	Fragmentation and Encryption . . . . .	167
8.4.1	Fragmentation Model . . . . .	168
8.4.2	Minimal Fragmentation . . . . .	169
8.4.3	Query Evaluation . . . . .	170
8.5	Departing from Encryption . . . . .	171
8.5.1	Fragmentation Model . . . . .	172
8.5.2	Minimal Fragmentation . . . . .	172
8.5.3	Query Evaluation . . . . .	174
8.6	Preserving Utility in Data Publication . . . . .	175
8.6.1	Visibility Requirements . . . . .	175
8.6.2	Loose Associations . . . . .	176
8.7	Conclusions . . . . .	179
<b>9</b>	<b>Selective Exchange of Confidential Data in the Outsourcing Scenario . . . . .</b>	<b>181</b>
	Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati	
9.1	Introduction . . . . .	181
9.2	Preliminaries . . . . .	183
9.3	Encryption Schema . . . . .	184
9.3.1	Key Agreement . . . . .	184
9.3.2	Key Derivation . . . . .	185
9.3.3	Encryption Policy . . . . .	187
9.4	Resource Sharing Management . . . . .	189
9.5	Comparison with the PGP's Key-Management Strategy . . . . .	191
9.6	Exposure Evaluation . . . . .	192
9.6.1	Anonymous Accesses . . . . .	192
9.7	Encryption Policy Updates . . . . .	194
9.7.1	Two-Layered Encryption Model . . . . .	195
9.7.2	Over-Encryption . . . . .	196

9.7.3	Collusion Evaluation . . . . .	196
9.8	Conclusions . . . . .	198
<b>References Part II . . . . .</b>		<b>199</b>
 <b>Part III Human Computer Interaction (HCI)</b>		
<b>10</b>	<b>PET-USES . . . . .</b>	<b>213</b>
	Erik Wästlund and Peter Wolkerstorfer	
10.1	Introduction . . . . .	213
10.2	PET-USES in Practice . . . . .	215
10.2.1	When to use the PET-USES . . . . .	216
10.2.2	How to use the PET-USES . . . . .	216
10.3	Conclusions . . . . .	217
10.4	Appendix: PET-USES[1.0] . . . . .	217
10.4.1	Instructions . . . . .	217
<b>11</b>	<b>HCI for PrimeLife Prototypes . . . . .</b>	<b>221</b>
	Cornelia Graf, Peter Wolkerstorfer, Christina Hochleitner, Erik Wästlund, and Manfred Tscheligi	
11.1	Introduction . . . . .	221
11.2	Overview of HCI challenges . . . . .	222
11.2.1	Challenge 1: Limited User Knowledge of PETs . . . . .	222
11.2.2	Challenge 2: Technologically Driven Development of PETs . . . . .	223
11.2.3	Challenge 3: Understanding PET Related Terms . . . . .	223
11.2.4	Challenge 4: Wrong Mental Models of PETs . . . . .	223
11.2.5	Challenge 5: Privacy as a Secondary Task . . . . .	224
11.2.6	Challenge 6: Complex Mechanisms are Hard to Understand . . . . .	225
11.3	Tackling the Challenges . . . . .	225
11.3.1	Limited User Knowledge of PETs . . . . .	225
11.3.2	Technologically Driven Development of PETs . . . . .	226
11.3.3	Understanding of PET Related Terms . . . . .	226
11.3.4	Wrong Mental Models of PETs . . . . .	227
11.3.5	Privacy as a Secondary Task . . . . .	227
11.3.6	Complex Mechanisms are Hard to Understand . . . . .	228
11.4	HCI Activities and Software Development . . . . .	228
11.4.1	Backup Prototype . . . . .	228
11.4.2	Privacy Dashboard . . . . .	229
11.4.3	Examples Reflected . . . . .	230
11.5	Discussion and Outlook . . . . .	231

<b>12 The Users' Mental Models' Effect on their Comprehension of Anonymous Credentials</b>	233
Erik Wästlund and Simone Fischer-Hübner	
12.1 Introduction	233
12.1.1 Anonymous Credentials	234
12.1.2 Related Work	235
12.2 Performed User Tests	236
12.2.1 Method	236
12.2.2 The Card-Based Approach	238
12.2.3 The Attribute-Based Approach	240
12.2.4 Results of the User Studies	242
12.3 Conclusions & Future Work	242
12.4 Acknowledgments	243
<b>13 Trust and Assurance HCI</b>	245
Simone Fischer-Hübner, Hans Hedbom, and Erik Wästlund	
13.1 Introduction	245
13.2 Social Trust Factors	246
13.3 A Trust Evaluation Function	247
13.3.1 Trust Parameters Used	247
13.3.2 Design Principles and Test Results	249
13.3.3 Test Results	251
13.4 The Data Track	253
13.4.1 Use of the Data Track	254
13.4.2 Test Scenarios & Test Setups	256
13.4.3 Results of the Usability Tests	257
13.4.4 Discussion of Data Track Usability Tests	259
13.5 Conclusions	260
<b>14 HCI for Policy Display and Administration</b>	261
Julio Angulo, Simone Fischer-Hübner, Tobias Pulls, and Ulrich König	
14.1 Introduction	261
14.2 Related Work	263
14.3 User Interfaces for Policy Management and Display	265
14.3.1 Selecting Privacy Preferences	266
14.3.2 The "Send Data?" Dialog	267
14.3.3 Testing the Usability of the "Send Data?" Dialog	273
14.4 Conclusions and Outlook	275
<b>15 Privacy Policy Icons</b>	279
Leif-Erik Holtz, Harald Zwingelberg, and Marit Hansen	
15.1 Introduction	279
15.2 Motivation for Introducing Privacy Icons	280
15.3 Related Work	280
15.4 PrimeLife Icon Sets	281
15.4.1 PrimeLife Icon Set for General Usage	281

15.4.2	PrimeLife Icon Set for Social Networks . . . . .	282
15.5	Test Results . . . . .	282
15.6	An Approach for Handling E-mail Data: Privicons . . . . .	284
15.7	Conclusions and Outlook . . . . .	285
<b>References Part III . . . . .</b>		<b>287</b>
 <b>Part IV Policy Languages</b>		
<b>16</b>	<b>Policy Requirements and State of the Art . . . . .</b>	<b>295</b>
	Carine Bournez and Claudio A. Ardagna	
16.1	Definitions . . . . .	295
16.1.1	Data Handling Policies . . . . .	295
16.1.2	Access Control Policies . . . . .	296
16.1.3	Trust Policies . . . . .	296
16.2	Legal Requirements . . . . .	297
16.3	Policy Language Requirements . . . . .	299
16.3.1	General Design Principles and Expressivity . . . . .	299
16.3.2	Requirements for Data Handling Policies . . . . .	300
16.3.3	Requirements for Access Control policies . . . . .	303
16.3.4	Requirements for Trust policies . . . . .	305
16.3.5	Other Technical Requirements for PrimeLife . . . . .	307
16.4	State of the Art . . . . .	308
16.4.1	Access Control Policy Languages . . . . .	308
16.4.2	Data Handling Policy Languages . . . . .	309
16.4.3	Anonymous Credential Systems and Private Information Management . . . . .	310
<b>17</b>	<b>Matching Privacy Policies and Preferences: Access Control, Obligations, Authorisations, and Downstream Usage . . . . .</b>	<b>313</b>
	Laurent Bussard, Gregory Neven, and Franz-Stefan Preiss	
17.1	Privacy Specifications: Preferences, Policies, and Sticky Policies . . . . .	313
17.2	Matching Data Handling . . . . .	315
17.2.1	Boolean Match . . . . .	315
17.2.2	Going Further than Boolean Match . . . . .	316
17.3	Obligations . . . . .	317
17.3.1	Triggers . . . . .	318
17.3.2	Actions . . . . .	319
17.3.3	Enforcement . . . . .	320
17.4	Authorisations . . . . .	321
17.5	Downstream Data Handling . . . . .	321
17.5.1	Structure of Downstream Authorisations . . . . .	322
17.5.2	Proactive Matching of Downstream Data Handling . . . . .	323
17.5.3	Lazy Matching of Downstream Data Handling . . . . .	324
17.6	Conclusion . . . . .	326



<b>18</b>	<b>Advances in Access Control Policies</b>	327
	Claudio A. Ardagna, Sabrina De Capitani di Vimercati, Gregory Neven, Stefano Paraboschi, Eros Pedrini, Franz-Stefan Preiss, Pierangela Samarati, and Mario Verdicchio	
18.1	Privacy-Preserving Access Control	327
18.1.1	Credentials Enabling Privacy-Preservation	328
18.1.2	A Policy Language for Privacy-Preserving Access Control	329
18.2	Credential Ontologies: Concepts and Relations	331
18.2.1	Abstractions	331
18.2.2	Delegation by Recursion	332
18.3	Dialog Management	333
18.3.1	Policy Sanitisation	334
18.4	Integration into XACML	336
18.4.1	Credential-Based XACML	338
18.4.2	SAML as Claims Language	340
18.4.3	XACML Architecture Extensions	340
18.5	Concluding Remarks	341
<b>19</b>	<b>Legal Policy Mechanisms</b>	343
	Leif-Erik Holtz and Jan Schallaböck	
19.1	Introduction	343
19.2	Legal Framework for Processing Personal Data	344
19.3	Gaps in Current Policy Language Approaches	346
19.3.1	XACML	346
19.3.2	P3P	347
19.4	Methodology	348
19.4.1	Looking into Privacy Policies	348
19.4.2	Looking at the Law	349
19.5	Use Cases	350
19.5.1	Online Shopping	350
19.5.2	Social Networking	352
19.6	Results and Further Research	353
<b>20</b>	<b>Policy Implementation in XACML</b>	355
	Slim Trabelsi and Akram Njeh	
20.1	Introduction	355
20.2	Architecture	356
20.2.1	High Level Architecture	356
20.2.2	Detailed Architecture	357
20.3	PPL Policy Language Structure	360
20.3.1	PolicySets, Policy and Rules	361
20.3.2	Credential Requirements	361
20.3.3	Provisional Actions	362
20.3.4	Data Handling Policies	362
20.3.5	Data Handling Preferences	363
20.3.6	Sticky Policies	363

20.3.7	Obligations . . . . .	364
20.3.8	Authorisations . . . . .	365
20.4	PPL Engine Data Model . . . . .	365
20.4.1	Package pii . . . . .	366
20.4.2	Package policy.Impl . . . . .	367
20.4.3	Package Credential . . . . .	369
20.4.4	Package Obligations . . . . .	371
20.4.5	Package StickyPolicy . . . . .	372
20.5	Conclusion . . . . .	374
<b>References Part IV . . . . .</b>		<b>375</b>
<b>Part V Infrastructures for Privacy and Identity Management</b>		
<b>21</b>	<b>Privacy for Service Oriented Architectures . . . . .</b>	<b>383</b>
Ulrich Pinsdorf, Laurent Bussard, Sebastian Meissner, Jan Schallaböck, and Stuart Short		
21.1	Introduction . . . . .	383
21.2	Requirements for Privacy in SOA . . . . .	385
21.2.1	Core Policy Requirements . . . . .	386
21.2.2	Privacy Logging Requirements . . . . .	387
21.2.3	Requirements for Access to Personal Information . . . . .	389
21.2.4	Cross-Domain-Specific Requirements . . . . .	389
21.2.5	Requirements for Additional Mechanisms . . . . .	390
21.3	Abstract Framework Addressing the Lifecycle of Privacy Policies in SOAs . . . . .	392
21.3.1	Privacy Issues Arising from SOA . . . . .	394
21.3.2	Abstract Protocol . . . . .	395
21.3.3	PII Provider . . . . .	398
21.3.4	PII Consumer . . . . .	400
21.3.5	Matching Abstract Framework with SOA Requirements . . . . .	402
21.4	Policy Composition . . . . .	404
21.4.1	Policy Composition Scenario . . . . .	405
21.4.2	Privacy Policy Composition Challenges . . . . .	406
21.4.3	Data-Centric Architecture for Privacy Enforcement . . . . .	408
21.4.4	Conclusion . . . . .	410
21.5	Outlook and Open Issues . . . . .	411
<b>22</b>	<b>Privacy and Identity Management on Mobile Devices: Emerging Technologies and Future Directions for Innovation . . . . .</b>	<b>413</b>
Marc-Michael Bergfeld and Stephan Spitz		
22.1	The Status: Privacy and Identity Management on Smart Mobile Devices . . . . .	413
22.2	The Changing Context (I): Multiple Partial Identities across Devices . . . . .	414

22.3	The Changing Context (II): Multiple Identity Providing Stakeholders Along an Increasingly Dynamic Mobile Services Value Chain .....	415
22.4	Technologies for Identity Management and Privacy Enhancement: Secure Elements .....	417
22.5	Present Secure Element Technologies: UICCs and Stickers .....	420
22.5.1	The Universal Integrated Circuit Card (UICC) and the Smart Card Web Server .....	420
22.5.2	The Sticker as Example for Static Mobile Service Identities .....	421
22.6	Emerging Secure Element Technologies: Trusted Execution Environments and the Privacy Challenge .....	422
22.7	Technologies for Secure and Dynamic Mobile Services and the Privacy Challenge in Highly Dynamic Environments .....	424
22.8	Contributions of the PrimeLife Project for the Advancement of Technologies in the Field .....	426
22.9	The Privacy Challenge in Mobile Services and Future Directions for Innovation .....	428
<b>23</b>	<b>Privacy by Sustainable Identity Management Enablers .....</b>	<b>431</b>
	Sascha Koschinat, Gökhan Bal, Christian Weber, and Kai Rannenberg	
23.1	Introduction .....	431
23.2	Economic Valuation Approach for Telco-Based Identity Management Enablers .....	432
23.2.1	Description of the Baseline Option and Feasible Delta Options .....	434
23.2.2	Identification of each Stakeholder's Costs and Benefits Based on Delta Scenarios in Comparison to the Baseline Scenario .....	436
23.2.3	Selection of Key Costs and Benefits for each Stakeholder	439
23.2.4	Mapping of each Stakeholder's Key Cost and Benefits on IdM Service Provider by Cause-Effect Chains .....	439
23.2.5	Clustering of Mapped IdM Service Provider Costs and Benefits .....	440
23.2.6	Assessment and Aggregation of Clustered IdM Service Provider costs and Benefits .....	443
23.2.7	Visualisation of Aggregated IdM Service Provider Costs and Benefits .....	445
23.3	Description of the Identity Management Scenarios .....	445
23.3.1	Authentication .....	446
23.3.2	Privacy Policy Enforcement .....	447
23.4	Related Work .....	451
23.5	Summary and Future Work .....	452
	<b>References Part V .....</b>	<b>453</b>

## Part VI Privacy Live

<b>24</b>	<b>Open Source Contributions</b>	459
	Jan Camenisch, Benjamin Kellermann, Stefan Köpsell, Stefano Paraboschi, Franz-Stefan Preiss, Stefanie Pötzsch, Dave Raggett, Pierangela Samarati, and Karel Wouters	
24.1	Introduction	459
24.2	Social Software	460
24.2.1	Clique – Privacy-Enhanced Social Network Platform	460
24.2.2	Scramble! – Audience Segregation by Encryption	461
24.2.3	Privacy-Awareness Support for Forum Users: Personal Data MOD	462
24.2.4	Privacy-Enhancing Selective Access Control for Forums	464
24.3	Dudle – <i>Privacy-enhanced</i> Web 2.0 Event Scheduling	464
24.4	The Privacy Dashboard	466
24.5	Privacy in Databases	470
24.5.1	Pri-Views – Protecting Sensitive Values by Fragmentation	470
24.5.2	Over-Encrypt	471
24.6	Anonymous Credentials	472
24.6.1	Identity Mixer Crypto Library	472
24.6.2	Components for a Privacy-Preserving Access Control System	473
24.7	Conclusion	474
<b>25</b>	<b>Contributions to Standardisation</b>	479
	Hans Hedbom, Jan Schallaböck, Rigo Wenning, and Marit Hansen	
25.1	Introduction	479
25.2	Standardisation in ISO/IEC JTC 1/SC 27/WG 5	480
25.2.1	ISO 24760 – Framework for Identity Management	481
25.2.2	Introducing Privacy Protection Goals to ISO 29101 Privacy Reference Architecture	482
25.3	Web Privacy	485
25.3.1	Workshop on Access Control Application Scenarios	486
25.3.2	Workshop on Privacy for Advanced Web APIs	488
25.3.3	Workshop on Privacy and Data Usage Control	489
25.3.4	Workshop on Internet Privacy	490
25.4	PrimeLife's Contributions to Standardisation in IETF	491
25.5	Conclusion and Outlook	491
<b>26</b>	<b>Best Practice Solutions</b>	493
	Marit Hansen	
26.1	Introduction	493
26.2	Recommendations to Industry	493
26.2.1	Data Minimisation by Pseudonyms and Private Credentials	494
26.2.2	Improvement of Privacy Functionality in Social Media	494

26.2.3	Better Protection of the User's Privacy on the Web . . . . .	496
26.2.4	Better Information of Users on Privacy-Relevant Issues on the Web . . . . .	496
26.3	Recommendations to Policy Makers . . . . .	497
26.3.1	Clear Guidelines for System Developers and Data Controllers . . . . .	498
26.3.2	Incentives and Sanctions . . . . .	499
26.3.3	Development of Law . . . . .	499
<b>References Part VI . . . . .</b>		<b>503</b>
<b>27</b>	<b>PrimeLife's Legacy . . . . .</b> Jan Camenisch and Marit Hansen	<b>505</b>
<b>Index . . . . .</b>		<b>507</b>



# Introduction



This introduction of the book first discusses the need for privacy, PrimeLife's vision and goals. It then elaborates on what privacy and identity are and why protecting on-line privacy is so hard. It finally summarises the results of the PrimeLife project that are presented in the different parts of this book.



# Chapter 1

## PrimeLife

Andreas Pfitzmann, Katrin Borcea-Pfitzmann, and Jan Camenisch

### 1.1 Motivation

The Internet continues to be increasingly valuable to individuals, organisations and companies. Web usage for everyday tasks such as shopping, banking, and paying bills is increasing, and businesses and governments are delivering more services and information over the Internet. Users have embraced the Internet for social networking and substantial collaborative works have emerged including Open Source initiatives, collaborative editing of encyclopedias, and self-help groups. Indeed, much of the information that is implicitly exchanged when people meet in person is now exchanged electronically over the Internet.

Businesses have also recognised the collaborative potential of the Internet. Companies offer services for such efforts. Enabled by the ease and effectiveness of on-line collaboration, businesses are becoming virtualised and are adopting ad-hoc collaboration and data-sharing. Information technologies have become pervasive and affect new areas of our daily lives. For example, a number of countries have or are about to introduce electronic identity cards and drivers licenses. Furthermore, electronic ticketing and tolling systems are in place all over the world. With the increasing number of communications systems, directories, personal information managers and social networks, the notion of sharing, viewing, and managing identity information becomes an important part of every business and government. This issue is a fundamental concept and people will be forced to deal with it.

Underlying all of these systems are distinct trust models and diverse trust relationships. Users and businesses rely increasingly upon information they find on the Internet – often without knowing anything about the originating sources. Thus, as a central part of their daily interactions, businesses as well as individuals need to manage not only their identity information but also trust information to assess their communication partners. For the safe future of the digital society, the concepts of privacy-enhancing user-centric identity and trust management are central. These concepts distinguish themselves from other notions of identity and trust manage-



ment by insisting that the user – and not some authority – maintains control over “what, where, when, why, and to whom” her personal information is released. This notion enforces user consent, which requires that (a) the user’s view of any transaction corresponds to the actual transaction and that (b) the user agrees to the execution of the transaction. For example, before a user logs into her banking website, she is told that she must prove (digitally) her name and birth date, for what purpose, and how her data will be treated. Then the user can either agree to this transaction and proceed or abort before her data is released. In user-controlled identity management, the user may moreover choose from many identity providers and also move her information between them. Thereby, important components are mechanisms for protecting a user’s privacy and anonymity, and yet simultaneously holding the user accountable if she commits fraud.

In the FP 6 project PRIME, it was shown that privacy-enhancing user-controlled identity management is feasible in today’s Internet with prevalent commerce between organisations (business, government) and consumers. The PRIME project has built and demonstrated the corresponding technology and shown how it enables privacy protection for individual citizens. While this is sufficient for traditional server-client style transactions, the Internet has undergoing fundamental changes in multiple areas, many of which pose new challenges to privacy and identity management:

*Community Focus:* The Internet fosters on-line community building. The main interactions here are between the members of the communities. Organisations act as intermediaries that store and serve data that is generated by individuals. There is daily evidence that these communities suffer from privacy and trust problems. Also, it has become common practice in companies to search the Internet for information about job applicants, which includes self-help and social networks such as Facebook and Linked-in. Thus, mechanisms are required that allow users to establish mutual trust while retaining their privacy, on the one hand, and to control the dissemination of their personal information, on the other hand.

*Mashup Applications:* New Internet paradigms and technologies are emerging. One of the important trends is service composition by means of AJAX and mashup technologies (Web 2.0). The dynamic composition and the contents originating from different, often questionable sources, makes it virtually impossible to assess the trustworthiness of contents – still, we all increasingly rely upon information retrieved from the Internet. Although the need for an authentication and identification infrastructure for the emerging Internet to establish trust is widely recognised, it is completely unclear how to build it and let alone how privacy can be protected in this environment.

*Lifelong Storage Enabling Unlimited Collection:* Storage has become virtually unlimited and cheap. Throughout our lives, we engage with a wide variety of different communities and organisations and thereby use different roles. Our first interactions happen as children, assisted by our parents and teachers, later as grownups in various professional roles, as parents, and as elderly persons possibly again assisted by relatives or social workers. An increasing portion of these interactions is digitised and will be stored forever. The protection of our private

sphere and the management of our trust relations are paramount. Previous work mainly focused on transactional privacy, i.e., on privacy at a given moment in time, and it is unclear what mechanisms are required to ensure life-time protection.

Summing up, within the limits set by law, privacy, trust, and identity management should support the user in deciding who can access which of her attributes in which situations. Thus, identity management has to do with the entire lifespan of people. Identity management has to accompany us from birth to death (and beyond) and throughout all areas of our life, i.e. private life, business life and any mixture thereof. Identity management has to cover all means of interaction we use with our family, friends, colleagues, employers, and public administration. These interactions will increasingly be done through or even mediated by computer networks.

## 1.2 Vision and Objectives of the PrimeLife Project

We envision that users will be able to act and interact electronically in an easy and intuitive fashion while retaining control of their personal data throughout their life. Users might use a multitude of different means to interact with several partners employing a variety of platforms.

For instance, a user Alice might authenticate to an on-line service created by a mash-up. She automatically logs on using her laptop and later confirms a payment transaction for an electronic article using her mobile phone. Despite many potentially untrusted services collaborating in this mash-up, Alice is required and able to reveal only the minimally necessary information to establish mutual trust and conduct the transaction. No service will learn any personal information about Alice. Nevertheless, a merchant is guaranteed payment for the services.

If Alice now wants to release personal data in an on-line dating service or in a network such as Facebook or MySpace, we envision that her personal sphere is protected. Even in such ‘data-intensive’ scenarios, Alice can trust that given personal data will only be released to peers that Alice trusts to safeguard this data. This corresponds to the real world where Alice releases sensitive data only if she trusts that the recipient properly protects it. So for instance, a future employer should not be able to access Alice’s entries in on-line dating or self-help forums since no sufficient trust has been established.

When the PrimeLife project was conceived, a number of privacy enhancing technologies such as private credential schemes and attribute-based access control already existed. It was shown by the PRIME project that effective privacy-enhancing identity managements systems can be built from these technologies. However, on the one hand, these technologies were not applied yet. On the other hand, these technologies seem not to be applicable to the Web 2.0 paradigm where users want and need to provide lots of personal information and furthermore, do not take into account that people and their roles change over time. Therefore, PrimeLife set itself

three objectives: The PrimeLife project aimed to address all of these issues and set itself the following three objectives.

- Research and develop new concepts, approaches, and technologies to protect privacy for Web 2.0 applications, such as social networks and blogs, and for lifelong privacy protection and management.
- Make existing privacy enhancing technologies useable and improve the state of the art.
- Foster the adoption of privacy enhancing technologies by providing open source components and educational materials and by cooperations with standardization bodies and dedicated workshops.

This book reports on the results that PrimeLife has achieved during its three years. In the remaining of this chapter we first discuss what we mean by privacy and then how we believe privacy can be protected in the digital society that is currently being build. We conclude this chapter by summarizing the results described in the different parts of this book.

### 1.3 Defining Privacy

The term *privacy* has been discussed for decades by different people in different occasions yet having (slightly) different meanings in mind. It has become a buzzword used nearly inflationarily. Numerous research papers try to analyse perceptions of privacy and to establish definitions for privacy (e.g., [Mar03, Par83, Phi04]). It is incontestable that privacy aims at protecting the autonomy of people, in the first place. Accordingly, Westin coined a widely accepted definition as follows:

“Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.” [Wes67]

In [BPPB11], the authors discuss the different stages of the historical development of privacy understanding and dealing with issues of privacy, which are the basis for the following discussion: In order to enable people to protect their privacy, technical as well as legal foundations were, and still are, required to be laid. Consequently, means to enforce confidentiality, i.e., hiding private information of an individual from unauthorised others, as well as data minimisation, i.e., limitation of storage and processing of personal data, entered the spotlight of researchers, developers, and regulators. In last two decades, the requirement of data minimisation became part of European legal regulations, cf. the data minimisation principle (Directive 2002/58/EC) and the purpose binding principle (Art. 6 (1b), Directive 1995/46/EC).

These days, many Internet users extend their social lives into the digital part of the world. This, however, conflicts with the traditional approaches of protecting privacy, namely confidentiality and data minimisation, as socialising, i.e., connecting to, communicating, and collaborating with each other, always means revealing at

least some personal information about oneself. Consequently, keeping personal data confidential or minimising data is socially not acceptable in every situation.

With respect to privacy, this means that people need to constantly evaluate appropriate trade-offs between protecting information about themselves on the one hand, and releasing as much personal information as it is reasonable for socialising on the other hand. Such trade-offs have two essential characteristics: Firstly, they have to be decided on for and by every user *individually*, i.e., in order to take control of their personal information, all users need to determine for themselves how their privacy is established. Secondly, the trade-offs are *highly context-dependent*. In other words, the selection of particular personal information to be revealed to others is conditional on the actual situation in which (current) activities take place, i.e., it depends on variables such as:

- who are the nearby actors (users that are potential or current *interactors* of the reference user);
- what exactly is the to-be-performed activity of the user in question – the *reference user*
- time of activity;
- frequency of activities;
- more specific properties, e.g., namespace of a Wiki page etc.

As described within the motivational section (cf. Section 1.1), *privacy-enhancing identity management* is one of the essential means of coping with the challenges that recent developments in the IT field pose to the users' privacy (as discussed above: Communities, Mashup Applications, and Lifelong Data Storage). The PrimeLife project, as well as its predecessor PRIME, took up these challenges to research and develop solutions that enable users to retain control over their personal information. The special focus of the PrimeLife project, the results of which are being described in this book, is on ensuring that citizens, i.e., the general public, adopt developed technology covering lifelong privacy as well as privacy-enhancing identity management.

## 1.4 From Identity via Identity Management to Privacy by Identity Management

The concepts of identity and identity management deal with describing someone's personality. But, what are the *data subjects*<sup>1</sup> that can have an "identity"? Almost everyone has natural persons in mind when referring to identities. However, an identity could also point to a legal person, of a group of users, or even to a computing device such as a laptop or mobile phone. The latter is true when, e.g., a person (let's call him Bob) takes a phone with him all the time. In this case, if Bob would allow

---

<sup>1</sup> By data subjects we refer to entities being able to interact via communication infrastructures with other entities, i.e., natural and legal persons as well as devices used to represent them in interactions. Sometimes, even sets of persons are called data subjects.

others to use the location tracking service of his phone, they could track at which times he moves where. This little example shows very well the need for identity management, not only for users, but also for the computing devices that represent them or even act on their behalf.

### 1.4.1 Identity – What It Is

The authors of [PBP10] conducted a structured analysis on the concepts of identity and privacy-enhancing identity management. Though the overall aim of their paper is to display the specifics of lifelong privacy and what consequences that setting has for privacy-enhancing identity management (see also Section 1.5), the basics of the concepts of identity and privacy-enhancing identity management are comprehensively outlined. The following statements put that analysis into the context of this book.

The concept of identity is only vaguely clear to most people. It relates not only to *names*, which are easy to remember for human beings, but goes far beyond *identifiers*, which typically link an identity to a certain context and which usually are unique in that context. As an initial approach, the notion of identity is described as follows:

*Identity* is a set of attribute values related to one and the same data subject.

Specifics and properties of identity-related attributes will be discussed later (cf. Section 1.4.2.1). Nevertheless, we allude here to attribute values being determined either by the identity holder himself or by others. Considering the earlier mentioned control over personal information by the owner, it is essential to consider this difference: differentiating between the two kinds of attribute assignment is crucial with respect to the possibilities of privacy management one has.

Considering time aspects, we have to extend the above introduced definition. Accordingly, attribute values used to specify an identity may change over time. Since all values an identity-related attribute can take are essential to describe the identity of its data subject, it is necessary to add a timestamp to each attribute value for which that attribute value is valid.<sup>2</sup> And, following this train of thought, we can further state:

An *identity* as a set of attribute values valid at a particular time can stay the same or grow, but never shrink.

This is true both for a global observer as well as for each party (or a set of parties) interacting with the entity represented by the identity. Therefore, if an unauthorised entity (a potential adversary) has no access to the change history of each particular attribute, the fact whether a particular subset of attribute values of an entity is an

---

<sup>2</sup> A *valid* attribute value means that it is used to represent its holder in a given setting.

identity<sup>3</sup> or not may change over time. If the adversary has access to the change history of each particular attribute, any subset of attribute values forming an identity, which sufficiently identifies its holder within a set of data subjects, will form such an identity from the adversary's perspective irrespective how attribute values change.

Any reasonable adversary will not just try to figure out attribute values per se, but the points in time (or even the timeframes) they are valid (in). This is because such change histories enable linking data and, thereby allowing the adversary to infer further attribute values. Therefore, it may help to define each *attribute* in such a way that its value(s) cannot become invalid. For example, instead of the attribute *location* of a particular individual person, take the set of attributes *location at time x*. Depending on the inferences one is interested in, refining that set as a list ordered concerning *location* or *time* may be helpful.

#### 1.4.1.1 Partial Identities

Bearing in mind that identities usually grow over time and the probability of identification of the entity within the given subset of entities usually grows as well, a solution is needed to solve this privacy-related dilemma, i.e., to preserve the entity's privacy. The idea is to subset the identity of an individual, the result of which should be a possibly very large set of so called *partial identities* [PH10]. Thereby, each partial identity may have its own name, own identifier, and own means of authentication. In a certain sense, each partial identity might be seen as a full-fledged identity of someone or something.

Selecting the appropriate approach for subsetting the attribute values is crucial, as it determines whether the establishment of partial identities is reasonable. Obviously, if subsetting is done poorly, e.g., by using identifying attributes within a major part of the entity's partial identities and, thus, allowing to link different partial identities to one and the same entity, it will not solve the privacy-related dilemma and it only makes the life of the respective person more complicated. Consequently, the right tools have to be used and subsetting one's identity has to be done in the right way. This does not only help the person whose identity is under consideration, but also the people communicating with her or him. That is because partial identities should consist of only those attribute values which are really needed within that particular relationship or context.

Figure 1.1 shows a snapshot of a person's ("John") possible partial identities in different contexts. The dark-grey areas represent different partial identities of John overlapping with parts of his full identity, represented by the light-grey area. While one may assume that this full identity, as well as its partial identities, are related to John's activities in either the online sphere or the physical sphere, activities may also spread to the other sphere.

<sup>3</sup> According to [PH10], an identity is defined as "any subset of attribute values of an individual person which sufficiently identifies this individual person within any set of persons" or more explicitly: "An identity is any subset of attribute values of an individual person which sufficiently distinguishes this individual person from all other persons within any set of persons."

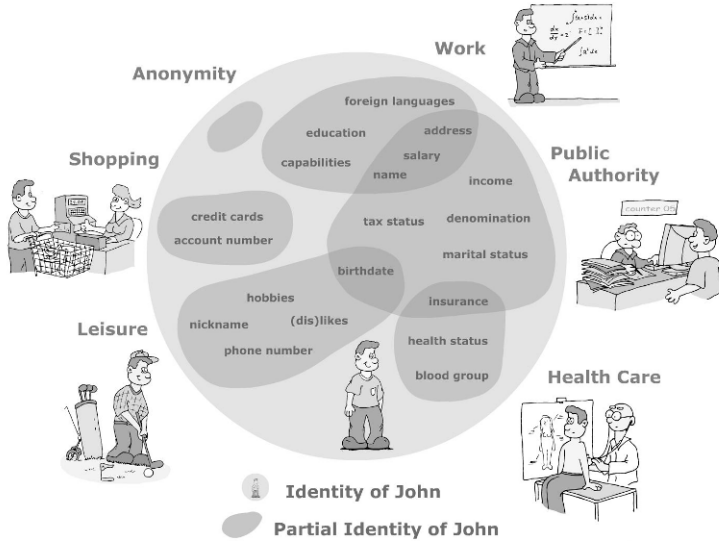


Fig. 1.1: Partial identities of an individual (“John”) [HBPP05].

The authors even assume that it is difficult to say if there will be any differentiation between those two “spheres” in the next 50 or 100 years. Ambient intelligence and ubiquitous/pervasive computing might make the boundaries blur and eventually disappear. This means that differentiating between identity-related data of the on-line and of the physical spheres might not make sense anymore. To conclude, when looking into the future, subsetting the identity/ies is absolutely essential whenever one strives for privacy.

#### 1.4.1.2 Requirements for Using Partial Identities

Taking advantage of partial identities requires a *basic understanding* of that concept by the data subject concerned. Of course, communication partners such as governments and businesses have to understand it as well, since managing one’s (partial) identities makes sense only if the interacting entities are willing to accept it.

Further, the authors assume that every person possess at least one *computer* (or some other device able to execute the according computations) that is administrating the person’s personal data and executing cryptographic protocols. Thereby, this personal computer is fully controlled by the user (otherwise there is no way to validate privacy properties).<sup>4</sup>

<sup>4</sup> This is in contrast to the typical digital rights management (DRM) scenario where the users have very limited control over their devices and the data processed by them. The authors are fully aware

By having a large set of (partial) identities, each of these (partial) identities needs its own means for secure authentication (otherwise there is no way to achieve accountability). This can be fulfilled by *digital pseudonyms*, which are similar to traditional (cryptographic) public keys and offer the strong privacy properties that we advocate. Besides allowing for secure authentication, digital pseudonyms represent unique identifiers of the respective (partial) identity and which are used to authenticate (sign) items originated by the holder in a way that recipients can check it (based on [PH10]).

Last but not least, means are required that enable a person to transfer certified attributes between her different partial identities. Therefore, it is important that this process by itself does not reveal that the different partial identities relate to the same person (unless the shared attributes uniquely identify the person). This transfer of certified attributes can be achieved by *anonymous credentials* which has been introduced by David Chaum in [Cha85] and a number of practical implementations are known today (cf. Chapter 5).

Indeed, anonymous credentials represent the appropriate basis for sharing certified attributes between partial identities of the same entity. Without anonymous credentials, the applicability of partial identities would be severely reduced.

### 1.4.2 Presentation of Identities – Pseudonyms

Considering the use of partial identities in particular, one has to be aware that, first, partial identities have to be consciously created and established. Secondly, usage patterns of the partial identities<sup>5</sup> drive linkability of the attribute values and, thus, the conclusions that could be inferred. This means that users should partition their online activities – a systematic approach of partitioning according to (disclosure) contexts of activity is called *context management* [BPPB11].

Identities or partial identities of an entity are represented using (*digital*) *pseudonyms*. Those pseudonyms serve as identifiers of the (partial) identities, on the one hand, and as addresses of the (partial) identities, on the other hand. In order to indicate holdership of a (partial) identity, an explicit link between the pseudonym and the holder of the attributes of that (partial) identity has to be created. Different kinds of initial linking between a pseudonym and its holder can be distinguished [PH10]:

---

that assuming that everyone has a computer fully under their control today is a very daring statement. However, when people talk about secure e-commerce, they assume the same. So, as there are “major commercial forces” striving for that direction, it could be expected that the assumption the authors have made will become a more realistic one during the next 20 years.

<sup>5</sup> When referring to *usage patterns of partial identities*, we address different aspects, e.g., how frequently a partial identity is communicated; how fine-grained is the context defined in which the partial identity is used; what rules are applied when selecting a particular partial identity for an interaction.



- *Public pseudonym*: The linking between a pseudonym and its holder may be publicly known from the very beginning, e.g., the phone number with its holder listed in public directories.
- *Initially non-public pseudonym*: The linking between a pseudonym and its holder may be known by certain parties (trustees for the according identity), but it is not public initially, e.g., a bank account with the bank as trustee for the identity.
- *Initially unlinked pseudonym*: The linking between a pseudonym and its holder is – at least initially – not known to anybody (except to the holder), e.g., biometric characteristics such as DNA (as long as it is not stored in a DNA register).

As previously mentioned, according to the usage patterns of partial identities and their pseudonyms, various types of pseudonyms can be distinguished. The differentiation of pseudonyms is closely related to different levels of anonymity that are achievable by the usage patterns.

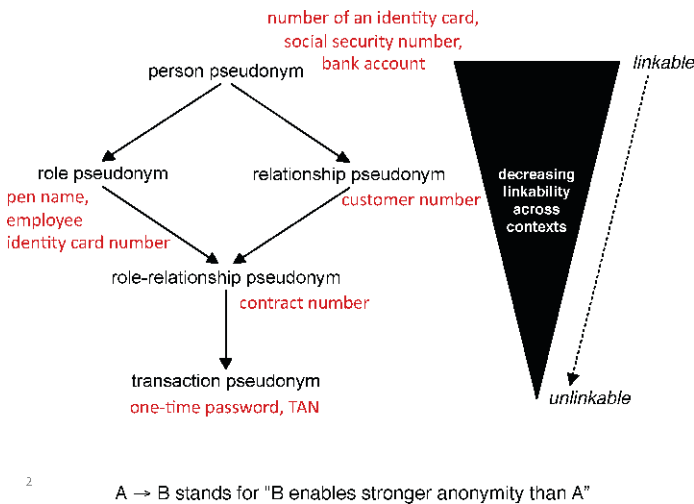


Fig. 1.2: Pseudonyms – Use in different contexts leading to partial order (based on [PH10]).

Figure 1.2 illustrates that interrelation. According to this, *person pseudonyms*, i.e., names or identifiers directly identifying a real person, imply the highest degree of linkability and, thus, they offer the least-possible anonymity to their holders. Examples for such kinds of pseudonyms are numbers of identity cards or the well-known social security number, which are used with very diverse communication partners and in very manifold contexts. Further, they typically are associated with their holders over their entire lifetime. This means, each time a user communicates by indicating her/his person pseudonym, all of that person's activities could poten-

tially be linked together. As a result, quite a detailed profile describing that person could be created.

In comparison, *role pseudonyms* and *relationship pseudonyms* are pseudonyms used within particular contexts only. Thereby, a role pseudonym is used by its holder when acting in a certain role. An example for role pseudonyms are pen names. Similar to role pseudonyms are relationship pseudonyms. They refer to entities within particular relationships, e.g., a pseudonym denoting someone in his or her relationship to a sports club. In this case, it does not matter if the person represents him- or herself in either of the roles, trainer or athlete. So, the two pseudonym types are distinguished according to the following rules: Whenever a pseudonym specifies a person communicating *with* specified other entities, then we speak of a relationship pseudonym. Instead of this, whenever users specify *as what/whom* they communicate, they are using role pseudonyms. Linkability is, therefore, restricted to the activities performed within the given relationship or when acting in a particular role and using the relevant pseudonym.

Even more privacy in terms of less linkability and stronger anonymity can be reached with help of *role-relationship pseudonyms*. The increase of conditions used in a particular relationship while appearing in a special role (e.g., appearing in the relationship to a particular sports club and either in the role as a trainer or as an athlete), narrows the variety of a scenario where one and the same pseudonym is used. In order to preserve privacy and to enable long-term communication with others, more role-relationships (and partial identities) have to be created for more specific contexts.

If the goal is to have the least linkability and utmost anonymity when communicating via a computer network, one has to make use of *transaction pseudonyms*, implying one-time use of pseudonyms. Linkability of different actions of the pseudonym holder via the pseudonyms only is not possible any longer since the user would create a new pseudonym for each interaction that is visible outside the user's personal computer.

The classification of pseudonyms as given above is a rather rough means to contribute to tool development supporting the user in decision making with respect to the selection of pseudonyms or partial identities. If a user decides, e.g., to re-use a pseudonym that initially was created to be used only once (i.e., for one transaction only), it will lose its property of a transaction pseudonym.

#### 1.4.2.1 Important Kinds of Attributes

When looking at attributes of (partial) identities, we can observe several kinds of attributes, each of them requiring a particular degree of protection when striving for privacy. In addition to the already mentioned attribute types name, identifier, and means of authentication, we distinguish biometrics, addresses (used for communication), bank accounts, credit card numbers etc. To a large degree, all of these are used for uniquely identifying entities. Biometrics as one of these attribute types has represented a well-known concept of the physical world used for identifying

persons for hundreds of years. However, biometrics being stored and evaluated by computers is relatively new. Biometrics can be helpful to bind computing devices to a natural person. But, it can also be critical if it is used in contradiction to privacy attitudes of people.

With respect to classifying identity-related attributes, there are different possibilities:

- One of the main distinctions that can be made with respect to attributes is if they are *authenticated*. If so, there are two possibilities regarding who authenticated the attribute: The first option is that they are authenticated by the first party – the data subject. In this case, it would be a claim that the data subject makes about her/himself and the claim would be as trustworthy as the data subject is trustworthy. The second option refers to authentication by a third party. Here, the authors explicitly do not refer to a *trusted* third party since the following two questions are to be clarified for each situation individually: Is the third party trusted by *whom* and with respect to *what*?
- Another approach of classification refers to *who knows* the attribute value, i.e., is the attribute value known only to the first party (the data subject) or also to second parties, i.e., the data subject's communication partners, or even to a third party that the first and second parties might not be aware of?
- Attributes can be classified according to the *degree of changeability*. Could attribute values be changed easily or is this hard to do? What possibilities does the entity have to change the attribute value?
- *Variability* of attributes over time is also a possible classification whereby this could range from non-varying to fully varying. In this context, it may also be interesting whether changes of attribute values with respect to when and what can be predicted?
- Attributes can be distinguished according to *who defines* the attribute values, i.e., are the attribute values given to the data subject by an external source or did the data subject her/himself choose the attribute values? This difference plays a special role in discussions of the possibilities of user control (cf. Section 1.4.1).<sup>6</sup>
- Further classification of attributes could be the actual *information* the attribute value contains. So, are we talking about *pure* attributes, whereby the attribute values contain only information about themselves, or do the attribute values also contain significant side information?<sup>7</sup>
- Also, attributes can be classified according to the *relationships* the data subject has. One could ask if an attribute value characterises a single entity per se or an entity only in its relationship to other entities, e.g., entity A likes/loves/hates entity B.
- *Sensitivity* of attribute values in certain contexts can be seen as an additional means to classify attributes, though this might be a very subjective approach.

<sup>6</sup> To give an example: if we refer to the attribute *colour of hair*, then its value can be a given (natural hair colour) or a chosen (after chemical dyeing) attribute.

<sup>7</sup> Let us assume we use biometrics, i.e., an image of someone's face available in a high resolution. From this, some doctors possibly may conclude some diseases.

However, if considering long-term use of attributes, then attributes judged to be non-sensitive today may become quite sensitive in future times (such as a possible change of the social order).

From those approaches of classification, conclusions can be drawn regarding how much protection attributes or attribute values need. Supposedly, some attribute values need much more privacy protection than others, e.g., those which

- are not easy to change,<sup>8</sup>
- do not vary over time or can be predicted,
- are given attribute values,
- might contain significant side information,<sup>9</sup> or
- are sensitive or might become sensitive in at least one context.

These attribute values are part of the *core identity*. Of course, it would be nice to protect everything. But, to be realistic, this is almost impossible – especially in situations where socialising is intended or even required. When one starts to manage identity attributes, one has to determine what defines her or his core identity: what attributes really belong to that core identity and need, therefore, relevant protection? Advancements and use of technology may shift some attributes from “core identity” to “non-core identity”; e.g., the address of someone’s house or flat is core for him/her, the current address of your laptop may not be.

### 1.4.3 Time Aspects of Identity Management and Privacy

Another interesting aspect one should consider when dealing with privacy issues is time-related aspects, which will be given as a first overview in the following subsection and in more detail in Chapter 4.

The design of privacy-preserving solutions and especially those aiming at privacy-enhancing identity management must not stop at supporting the user in managing her/his *present* identities. Instead, since any kind of privacy intrusion may have implications on the individual’s future life, it is necessary that the issues related to longterm aspects of privacy-enhancing identity management are identified and understood.

Controlling the disclosure of one’s personal data throughout his/her entire life comprises a timeframe of nearly 100 years and, seen from the current moment of time, it takes the past, the present, and the future of the person concerned into account. During that timeframe, an individual’s world can change drastically, i.e.,

---

<sup>8</sup> To give an example of the necessity to protect those attributes, imagine biometrics becoming widely known. Then, it might become necessary, but be very hard, to change the biometrics (which could mean, e.g., handing out new fingerprints to everybody). In comparison, cryptographic keys can easily be revoked and new ones generated.

<sup>9</sup> Nobody knows which algorithms for analysis of side information will become available during the next years.

information and communication technology develops and each individual's appreciation of privacy will change several times in her or his life.

In the authors' opinion, it is difficult, if not impossible, to make data fade away. Each time a user uses the Internet, s/he creates traces. What s/he cannot do is reliably cause data to be destroyed on other persons' or organisations' machines. This is a very important issue to be considered in this context. Accordingly, we need mechanisms that can realise the above mentioned privacy-related concepts. In the first place, *hiding* should be given priority over disclosing data. For this, identity management and user control are the right means. Also, it is essential to have assured long-term security by using information-theoretically secure cryptography [PBP10].

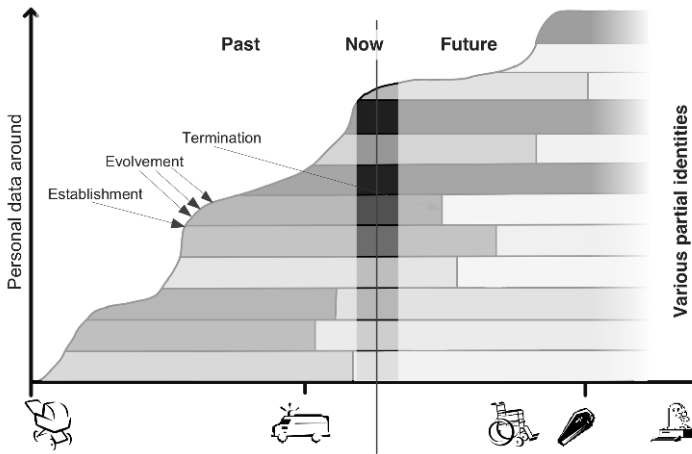


Fig. 1.3: Example of how partial identities develop throughout life and in various areas of life (based on [CHP<sup>+</sup>09]).

As indicated, a person's attitude regarding his/her privacy will change as s/he runs through various *phases of life* and acts in different *areas of life*. Figure 1.3 is an attempt to depict disclosures of personal data during an individual's lifetime, which has been sketched in [CHP<sup>+</sup>09, HPS08]. Usually, even before a human being is born, a lot of personal data about the unborn child is gathered. Such gathering continues during a human being's life. The data is stored with various data controllers involved as individual partial identities. Finally, when the person passes away, the evolution of the (partial) identities is terminated. But, termination does not mean that the data disappears. In many cases, the data will be stored further, e.g., in back-ups.

To conclude, the management of lifelong privacy means (1) covering the *full lifespan* by considering short-term as well as long-term effects; (2) covering *all*

*areas of life* by addressing context-specific as well as context-spanning aspects; (3) covering *different stages of life* by respecting constant as well as changing abilities or behaviour of individuals.

## 1.5 Further Facets of Privacy

Besides the basic concepts and ideas related to privacy and discussed in this chapter, privacy combines further aspects. A selection especially interesting in the context of the PrimeLife project will be explored in the following.

Initially, we indicated that people have different understandings of the concept privacy whereby this actual fact primarily can be attributed to the historical development of privacy perceptions or to focal points of work or social lives of people.<sup>10</sup> The following list indicates main concepts related to privacy, which are sometimes put on the same level as privacy (note that these items are not to be understood as being on equal footing):

- Related to direct identifiability (the concepts have a direct relation to the identities and identifiability of persons):
  - *Anonymity* in the meaning of non-identifiability of an entity within a certain set of entities [PH10];
  - *Pseudonymity* in the sense of being recognisable in selected situations while (re-)using particular pseudonyms.
- Related to indirect identifiability (the concepts relate to the personal data of entities possibly used to (not) identify the entity concerned):
  - *Confidentiality* – hiding private things from unauthorised users by establishing a private sphere. Confidentiality can be achieved using cryptographic algorithms applied to the communicated data itself or to the communication channel;
  - *Data minimisation* primarily used within legal contexts refers to limiting the storage and processing of personal data: Data minimisation is, in the first place, a legal means to assure privacy, but it should also be one of the basic principles of application developers when designing social software, i.e., applications have to be designed in such a way that processing personal data needs to be kept to a minimum and users should be offered different options that allow for data minimisation;
  - *User control* relates to users who determine themselves which of their personal data they disclose to whom in which situation.
- *Contextual integrity* as a privacy denotation is not yet a common idea. Despite the term's novelty, it refers to two privacy-related understandings: The first was

---

<sup>10</sup> Often, even researchers equate privacy with, e.g., anonymity.

coined by Helen Nissenbaum in [Nis98]. In order to protect privacy in Nissenbaum's understanding, personal data must not leave the originating social context. This approach is similar to the already indicated ideas of minimising personal data. The second understanding has been discussed by Borcea-Pfitzmann et al in [BPPB11]. Their idea is to allow personal data to be distributed and processed – under the condition that data describing the context from where the personal data originates is transferred together with the actual personal data.

This list of privacy-related concepts is neither intended to be complete nor to rank those concepts. Instead, it should give the reader a insight into the different views on and approaches to privacy that researchers, developers and users have to deal with.

## 1.6 How to Protect Privacy and PrimeLife's Contributions

Protecting one's digital privacy involves foremost managing and controlling the release and dispersal of one's personal information. Of course, we cannot hope to completely control information about ourselves in an open society and, in particular, when using open communication media such as the Internet. Worse, today we have essentially lost control over our information, its use and dispersal by others, be it by corporations for profit, by other users for fun, or by criminals for identity fraud and other misconducts.

Regaining control over our personal data is not only important to protect ourselves from fraud but also it also required for our future society and marketplace to flourish: A democracy cannot work, e.g., without elections where citizens are guaranteed anonymous votes, and in markets where information itself becomes a good, it is essential that information be protected, and its use and dispersal can be governed by its owner.

While there is some legal and social protection of our privacy, the best way to protect our privacy is not to reveal any information about ourselves at all – which of course does not work: we needed to provide information about ourselves whenever we want to interact with other people or service providers or use applications over networks. We advocate, however, that the information we need to reveal in interactions be minimised, on the one hand, and, once revealed, that the information be protected and its use be governed by policies and access control systems, on the other hand. This requires electronic services and applications to be “privacy by design.” It is not sufficient that the designers and developers are privacy minded but they also need to know what privacy-enhancing technologies can offer, how it works, how it can be employed, and to be provided these technologies. Privacy-enhancing technologies include the classical ones such as onion routing protocols to anonymous communication, private credential systems and minimal disclosure tokens, and database anonymisation techniques. They also include suitable access control mechanisms and policy languages, infrastructure components and, most important, user interfaces that enable the users to execute their control.

The technologies known before PrimeLife started, primarily focused on single transactions, how to minimise the information released in these transactions, and how to protect this information once it was released. However, these approaches do not take into account the fact that people and their roles change over time, nor do they fit the paradigm shifts of the so-called Web 2.0, where people use the Internet quite differently and share lots of personal data on social networks.

As we have already mentioned, the PrimeLife project aimed to enable users to protect their privacy in a digital society by the following three objectives.

- Research and develop new concepts, approaches, and technologies to protect privacy for Web 2.0 applications, such as social networks, and collaborative applications, and for lifelong privacy protection and management.
- Make existing privacy enhancing technologies useable and improve the state of the art.
- Foster the adoption of privacy enhancing technologies by providing open source components and educational materials and by cooperations with standardisation bodies and dedicated workshops.

The work towards these goals was structured in six activities, each consisting of a number of work packages (WPs). The first activity (Privacy in Life) addressed the first objective and produced a number of new mechanisms to protect privacy in social networks and Internet forums. It also studied the problem of access to personal data and the delegation thereof.

Activities 4 (HCI), 5 (Policies), and 6 (Infrastructure) addressed the second objective. In particular, Activity 5 was concerned with developing policy languages and access control systems for attribute-based access control enabling the use of anonymous credentials or minimal disclosure tokens for authentication and hence privacy-enabling access control. Activity 6 studies the infrastructure requirements for privacy-enhancing identity management and access control and how one can have the infrastructure changed towards this. Activity 4 researched and developed user interfaces for the different mechanisms that the project developed so that they become usable by the end users.

Activity 2 (Mechanisms) could be seen as coming up with new mechanisms and improving the existing ones as needed by all the other activities and has indeed produced an impressive number of new research results and prototypes. Finally, Activity 3 (Privacy Live!) was concerned with fostering the adoption of the technologies produced by PrimeLife. To this end, a number of workshops were held, contributions to standardisation bodies made, and many of the technologies were provided as open source components. Also, the book you are holding in your hands is a result of this activity.

The different parts of this book present the results of the different activities of PrimeLife. The following section provides brief summaries of these results per activity.



### ***1.6.1 Part I - Privacy in Life***

It was a goal of PrimeLife to provide sustainable, scalable and configurable privacy and identity management for an individual in various situations of his life.

The first focus of the research in this area was on assisting users in new and emerging Internet services and applications such as virtual communities and Web 2.0 collaborative applications. This was achieved by setting up demonstrators showing how audience segregation for data can be done as documented in Part I, Chapter 2. Additionally in Chapter 3, we describe trust mechanisms to help users decide on trustworthiness in data delivered by others with all users concurrently having privacy requirements on the data they request and deliver. The scenarios we chose for Chapters 2 and 3 cover a broad bandwidth of Web 2.0 applications (blogs, wikis, social networks, forums). The prototypes we built for these scenarios served as a basis for experiments for finding out which indicators raise users' awareness with respect to data privacy and trustworthiness.

The second focus of research was on the life-time aspect of privacy a user faces as we will outline in Part I, Chapter 4. Each individual leaves a multitude of traces during a lifetime of digital interactions. While parts of these traces are unconsciously left behind and not meant as important information, lots of information is very relevant and important for specific domains. This information, containing information about the individual, but also all sorts of documents, has to be accessible by someone at any time. This implies that, in case an individual is not able to manage this information, others need to obtain access. As exemplary demonstrator, we built a backup tool that takes into account new ways of interacting. The tool provides for the backup and synchronisation of documents. With a view on the specific requirements concerning lifelong privacy and identity management, mechanisms are built in to safeguard individuals, to improve control over the data and to allow for delegation.

### ***1.6.2 Part II - Mechanisms to Enable Privacy***

Today's society places great demand on the dissemination and sharing of information. Such a great availability of data, together with the increase of the computational power available today, puts the privacy of individuals at great risk. The objective of the mechanisms activity is therefore to do novel research on the different open issues of the complex problem of guaranteeing privacy and trust in the electronic society. Chapter 5 focuses on privacy-enhancing cryptographic technologies that can be used in practice. The chapter presents anonymous credential schemas and their extensions along with cryptographic applications such as electronic voting and oblivious transfer with access control. Chapters 6 and 7 addresses mechanisms supporting the privacy of the users (transparency support tools, privacy measurement) and their electronic interactions. In particular, Chapter 6 illustrates a privacy-preserving secure log system as an example of a transparency supporting tool and an interoperable

reputation system. Chapter 8 investigates the problem of assessing the degree of protection offered by published data and of protecting privacy of large data collections that contain sensitive information about users. The chapter presents an information theoretic formulation of privacy risk measures and describes fragmentation-based techniques to protect sensitive data as well as sensitive associations. Chapter 9 addresses the problem of providing users with means to control access to their information when stored at external (possibly untrusted) parties, presenting new models and methods for the definition and enforcement of access control restrictions on user-generated data. The chapter illustrates a novel solution based on translating the access control policy regulating data into an equivalent encryption policy determining the keys with which data are encrypted for external storage. The solution is complemented by an approach based on two layers of encryption for delegating to the external server possible updates to the access control policy (without the need for the data owner to re-encrypt and re-upload resources).

### ***1.6.3 Part III - User Interfaces for Privacy Management***

Privacy-enhancing Identity Management will only be successful if its technologies are accepted and applied by the end users. For this reason, the research and development of user interfaces for PrimeLife technologies, which are intelligible, user-friendly while compliant with legal privacy principles, and which are mediating trust, have played an important role in the PrimeLife project and have been addressed by PrimeLife Activity 4 (HCI). The main research achievements of Activity 4 that will be reported in this book can be structured by the Activity 4 work packages:

The first three chapters of Part III of this book report the main research results on novel HCI methodologies for PETs, on mental models and metaphors for privacy-enhancing identity management and has helped to develop and evaluate UIs for PrimeLife prototypes. Chapter 10 reports on PET USES - the Privacy Enhancing Technology Self-Estimation Scale which we have developed and used within PrimeLife for evaluating user interfaces for privacy enhancing technologies. Chapter 11 discusses the HCI development process and testing of PrimeLife prototypes. Chapter 12 describes a series of mockups for anonymous credential selection based on a card metaphor and analyses what effects the users' mental models have on their understanding of the selective disclosure property of anonymous credentials. In particular, we investigate and compare the effects of the mental models of a card-based user interface approach and an attribute-based user interface approach.

Chapter 13 considers Trust and Assurance HCI and investigates UI mechanisms to provide and communicate trustworthiness of Privacy and Identity Management technology to the end users. For this, the iterative design process of a trust evaluation function is described, which allows end users to evaluate the trustworthiness of services sides in terms of their business reliability and their privacy practices. Transparency can be a means for enhancing the users' trust in PrimeLife technolo-

gies. Chapter 13 also describes the iterative user interface design of the data track, which is a user-friendly transparency tool consisting of a history function documenting what personal data the user has revealed under which conditions plus online functions allowing a user to exercise her right to access her data at remote services sides.

Our research on User Interfaces for Policy Display and Administration has elaborated user-friendly and legally compliant forms of privacy policy definition, administration, negotiation, and display. Chapter 14 presents our work on user interfaces for a simplified management of privacy preferences and for a user-friendly display of data handling policies of services sides including information about how far they match the user's privacy preferences. PrimeLife's work on privacy policy icons for presenting policy components in a very visible and intuitive manner is presented in Chapter 15.

#### ***1.6.4 Part IV - Policies to Control Privacy***

Machine-interpretable policy languages are a key part of any modern privacy infrastructure. PrimeLife set out to collect policy language requirements from the diverse scenarios covered by the project, which are summarised in Chapter 16. After an analysis of the suitability of existing policy languages, it quickly became clear that none of them covered all the needs we discovered.

The main highlight of the policy activity is the specification and implementation of the PrimeLife Policy Language (PPL), an integrated policy language allowing data controllers to express which data they need from data subjects and how this information will be treated, and at the same time allowing data subjects to express to whom and under which conditions they are willing to reveal their information. Chapter 17 focuses on the relation between access control policies and data handling policies, and describes an automated matching procedure by which means a proposed policy can be automatically matched against a data subject's preferences. Chapter 18 introduces privacy-friendly access control policies by proposing the concept of "cards" as a generalisation of several existing authentication technologies, including anonymous credentials. Chapter 20 reports on the architecture and implementation of the PPL engine that brings the advanced research concepts to life.

Chapter 19 takes a closer look at the legal requirement under European law to transparently inform users about the usage of their information. Expressing such usage in an understandable way is a notorious challenge. Faced with the multitude of applications and usage purposes and with the lack of a structured ontology among them, this chapter investigates the current practices in data usage in various contexts and discovers a common structure.

Finally, some of the most important concepts of the performed policy research, in particular the results presented in Chapters 17 and 18 were brought together in the design of the PrimeLife Policy Language (PPL). To be of use in real-world settings, PPL was defined as extensions to the industrial standards XACML and SAML.

### ***1.6.5 Part VI - Infrastructures for Privacy***

While infrastructure aspects have a significant impact on the adoption, security and privacy functionality of IdM systems in general, and of privacy enhancing identity management systems in particular, they are often overlooked. One reason is the complexity of the infrastructure aspects, as often every element has some relation to every other element of an infrastructure, and identity management infrastructures must be interoperable among themselves or with existing legacy solutions.

Part VI concentrates on the three most relevant aspects in infrastructures and infrastructure research:

1. Privacy for service-oriented architectures: How can privacy be integrated into service-oriented architectures (SOA) that define more and more aspects of the Internet-based business? Chapter 21 first lists legal and technical requirements for privacy in service-oriented architectures. These requirements form the starting point for a technical framework that brings privacy enhanced data handling to multi-layered, multi-domain service compositions. Further, an abstract framework is described that is technology agnostic and allows late adoption also in already existing SOA applications.
2. Smart mobile devices: Technologies and future directions for innovation. Chapter 22 elaborates upon the existing and upcoming technologies for an increasingly dynamic creation of services between front-end mobile devices and back-end servers and sketches how a conscious inclusion of security, identity-management and privacy-enhancement can be achieved in the future.
3. Privacy by sustainable identity management enablers: To optimise sustainability, an economic valuation approach for telco-based identity management enablers is developed. This will enable telecommunications operators to learn the relevant factors for assessing privacy-enhanced IdM enablers (Chapter 23). Together, these chapters address the roles that networks, (or network architectures), devices, and services play for infrastructures considering the interest of the respective stakeholders.

### ***1.6.6 Part VII - Privacy Live!***

One of the main objectives of PrimeLife was to bring the tools we developed to use and to make end-users and other stakeholders aware what privacy enhancing mechanisms can achieve. To this end, we have published a wide variety of code on the project's website for download. Most of them are open source in the classical sense and all of them can be used for free and the source code and other documentation is available. Chapter 24 describes a selection of the tools we have published; for the complete list we refer to PrimeLife's website.<sup>11</sup>

---

<sup>11</sup> <http://www.primelife.eu/>

As a second way to make privacy live, PrimeLife has contributed to standardisation organisations. The main focus here was on the relevant ISO working groups and the W3C-related bodies and our work is described in Chapter 25.

As a third way to improve privacy in the real world, we considered how existing technology is used for privacy- relevant data processing. In many cases, it neither matches the provisions of European data protection regulation nor does it address society's and individuals' needs for maintaining privacy throughout a full lifetime. To address this, we elaborated on requirements and recommendations for all stakeholder groups involved. We present a selection of our best practice solutions that address different stakeholders in Chapter 26.

Besides the results described in this book, we have further organised a number of summer schools. These schools had two facets. First, they brought together senior researchers and practitioners from many disciplines, most of them gave keynote talks on different aspects of privacy. Second, PhD students were invited to submit their research results and a selection of the research papers have been presented and discussed at the summer schools. The thereafter revised papers were published as proceedings [BDFHMH10].

Last but certainly not least, all PrimeLife partners have published extensively at scientific conferences. A selection of the results presented at these conferences and published in proceedings and journals are described in this book. For the complete publication list we (again) refer to PrimeLife's website.

# References Introduction

- [BDFHMH10] Michele Bezzi, Penny Duquenoy, Simone Fischer-Hübner, and Ge Zhang Marit Hansen, editors. *Privacy and Identity Management for Life*. IFIP Advances in Information and Communication Technology. Springer, Boston, 2010.
- [BPPB11] Katrin Borcea-Pfitzmann, Andreas Pfitzmann, and Manuela Berg. Privacy 3.0 := Data Minimization + User Control + Contextual Integrity. *it - Information Technology*, 53(1), January 2011. To be published.
- [Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
- [CHP<sup>+</sup>09] Sebastian Clauß, Marit Hansen, Andreas Pfitzmann, Maren Raguse, and Sandra Steinbrecher. Tackling the challenge of lifelong privacy. In *eChallenges*, October 2009.
- [HBPP05] Marit Hansen, Katrin Borcea-Pfitzmann, and Andreas Pfitzmann. PRIME – Ein europäisches Projekt für nutzerbestimmtes Identitätsmanagement. *it – Information Technology, Oldenbourg*, 6(47):352–359, 2005.
- [HPS08] Marit Hansen, Andreas Pfitzmann, and Sandra Steinbrecher. Identity management throughout one’s whole life. *Information Security Technical Report*, 13(2):83–94, 2008.
- [Mar03] S.T. Margulis. Privacy as a social issue and behavioral concept. *Journal of Social Issues*, 59(2):243–261, 2003.
- [Nis98] H. Nissenbaum. Protecting Privacy in an Information Age: The Problem of Privacy in Public. *Law and Philosophy*, 17(5):559–596, 1998.
- [Par83] W. A. Parent. A new definition of privacy for the law. *Law and Philosophy*, 2:305–338, 1983. 10.1007/BF00144949.
- [PBP10] Andreas Pfitzmann and Katrin Borcea-Pfitzmann. Lifelong Privacy: Privacy and Identity Management for Life. In Michele Bezzi, Penny Duquenoy, Simone Fischer-Hübner, Marit Hansen, and Ge Zhang, editors, *Privacy and Identity Management for Life*, volume 320/2010 of *IFIP Advances in Information and Communication Technology*, pages 1–17, Boston, 2010. Springer.
- [PH10] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management v. 0.34. [https://dud.inf.tu-dresden.de/Anon\\_Terminology.shtml](https://dud.inf.tu-dresden.de/Anon_Terminology.shtml), August 2010.
- [Phi04] David J. Phillips. Privacy policy and PETS – The influence of policy regimes on the development and social implications of privacy enhancing technologies. *New Media & Society*, 6(6):691 – 706, 2004.
- [Wes67] Alan F. Westin. *Privacy and Freedom*. New York Atheneum, 1967.



**Part I**  
**Privacy in Life**





## Introduction

New information technologies are very popular and successful because individuals want to share personal and non-personal data with others where ever they are. However, individuals in the Information Society want to safeguard their autonomy and retain control over their personal information, irrespective of their activities. This means that users of information technologies always have to be aware of which data they are producing explicitly and implicitly. They need to know who their audience is and they have to decide whether to trust the members of their audience or not. Information technologies need to transfer the measures for establishing trust we have in the offline world (e.g., usage of ID cards, certifications of rights, word-of-mouth, ...) to the Information Society. This raises additional privacy risks for individuals as the information about them for establishing trust needs to be linkable to them. Here, a paradox related to online interaction becomes apparent. On the one hand, online interaction means that more traces are left and that individuals can be recognised easier, while, on the other hand, the physical distance between interacting parties makes it more difficult to define exactly with whom one is interacting.

It was a first short-term goal of PrimeLife to provide scalable and configurable privacy and identity management in new and emerging Internet services and applications such as virtual communities and Web 2.0 collaborative applications. This was achieved by setting up demonstrators showing how audience segregation for data can be done as documented in Chapter 2. Additionally, in Chapter 3, we describe trust mechanisms to help users decide on trustworthiness in data delivered by others, with all users concurrently having privacy requirements on the data they request and deliver. The scenarios we chose for Chapters 2 and 3 cover a broad bandwidth of Web 2.0 applications (blogs, wikis, social networks, forums). The prototypes we built for these scenarios served as a basis for experiments for finding out which indicators raise users' awareness with respect to data's privacy and trustworthiness. Unfortunately, within PrimeLife, it was not possible to make large-scale experiments with a large public. But most of the tools we built for these scenarios are also available as open-source for further deployment and usage.

A second longer-term goal of PrimeLife as documented in Chapter 4 is to protect the privacy of individuals over their whole span of life. Each individual leaves a multitude of traces during a lifetime of digital interactions. The total of these traces forms a digital footprint. While part of these traces is unconsciously left behind and not meant as important information, a lot of information is very relevant and important for specific domains. This information, containing information about the individual, but also all sorts of documents, has to be accessible by someone at any time. This implies that, in case an individual is not able to manage this information, others need to obtain access. The inability to manage information can be either temporary, such as during the case of illness, or permanent, when the individual deceases. As an exemplary demonstrator, we built a backup tool that takes into account new ways of interacting. The tool provides for backup and synchronisation of documents. With a view on the specific requirements concerning privacy and identity

management, mechanisms are built in to safeguard individuals, to improve control over the data and to allow for delegation.

## Chapter 2

# Privacy in Social Software

Bibi van den Berg, Stefanie Pötzsch, Ronald Leenes, Katrin Borcea-Pfitzmann, and Filipe Beato

**Abstract** While using social software and interacting with others on the Internet, users share a lot of information about themselves. An important issue for these users is maintaining control over their own personal data and being aware to whom which data is disclosed. In this chapter, we present specific requirements and realised solutions to these problems for two different kinds of social software: social network sites and web forums.

### 2.1 Scenarios and Requirements

In recent years, a new generation of the Internet has emerged, also known as 'Web 2.0'. One often quoted definition of Web 2.0 states that this term refers to a "set of economic, social, and technological trends, that collectively form the basis of the next generation of the Internet – a more mature, distinct medium characterised by user participation, openness, and network effects" [MSF09]. Web 2.0 has four fundamental characteristics that set it apart from the first generation of the Internet ('Web 1.0'):

- Internet users have changed from passive consumers of information (searching for information and reading materials provided by others) into *active creators of content* [Tap09, How08, Lea08]. In Web 2.0, users can share their knowledge and information via a wide range of channels. Blogs, YouTube movies, wikis, file-sharing and consumer reviews are examples in case.
- In Web 2.0, social interaction plays a central role. This is why Web 2.0 is also called '*the social web*'.
- In many Web 2.0 environments, sharing and creating content and knowledge is not a solitary enterprise, but quite the reverse: the production and dissemination of information and entertainment services has a highly co-operative character. *Participation* and *co-creation* are key aspects of Web 2.0.

- Web 2.0 also differs from the first generation of the Internet in a technical sense: technology developers now create applications that are *embedded* into the Internet, and are accessible via any browser. Thus, the Internet has become the central platform for users to access different types of software [O’R07]. Moreover, software is offered to users as a service rather than as a product to buy separately.

Since Web 2.0 is a new phenomenon – the term was first coined in 1999 but the massive take-off of this latest generation of the Internet is only a few years old – much is still to be learned with regards to both the benefits and the risks for users, businesses and governments in this new domain. Privacy issues relating to modern technologies have been high on the agenda of both government officials around the world, researchers, and the broader public, and for good measure, since it is obvious that the emergence of Web 2.0 currently generates a wide range of new issues relating to privacy and security.

As said, the success of the social web is based on the active participation of users, and on their willingness to contribute to the creation and improvement of content on the Internet by sharing data and knowledge [SGL06, O’R07]. By using social software, a lot of personal data is disclosed either directly – think of real names and birth dates on social networking sites – or indirectly, for instance through editing specific topics in a wiki, commenting on blog entries or posting statements in a forum [GA05, EGH08]. Furthermore, personal data can be generated by establishing connections with, or disclosing information by, second parties with or without the consent of the respective person. While the possibilities of the social web may enrich people’s lives on the one hand, there are also privacy risks involved. Five central privacy issues can be distinguished with respect to information and communication technologies in general, and Web 2.0 applications in particular:

- When sharing data and knowledge in social software, users lack an overview of who has access to this information – they cannot adequately judge the size and makeup of their *audience* [PD03, Tuf08].
- Information and communication technologies enable anyone to collect, copy, link and distribute the (personal) data of others, thus allowing for the creation of extensive profiles of individual persons. Information may also easily be copied outside the original domain, thus making it even harder for users to know who has access to their information [Hou09].
- Information and communication technologies allow storage of data for a nearly indefinite time period, thus making it impossible to erase or forget this information [MS09].
- Participatory information and communication technologies such as social software enable anyone to publish another individual’s personal data, which may have serious consequences for the other’s reputation [Sol07].
- Individuals’ lack of privacy-awareness when using social software may lead to information leaks and leaving unintended and/or unobserved virtual traces.

To find out which guises privacy issues take in the new generation of the Internet, much research has been conducted with regards to privacy in social software in

recent years, and especially in social network sites. One of the central points with regards to information sharing and privacy in social network sites is aptly summarised by Acquisti and Gross. They write: ‘...one cannot help but marvel at the nature, amount, and detail of the personal information some users provide, and ponder how informed this information sharing is’ [AG06]. Individuals use social networking sites for two main goals: (1) to *present themselves* to others in a virtual domain, and (2) to engage in, manage and strengthen *relationships*. Both of these involve several privacy risks as Scenario 1 (Section 2.1.1) will show.

A much less researched, yet highly relevant domain for privacy issues in Web 2.0 is that of *collaborative workspaces* and *forums*. Users primarily use these environments to create and share *content*, rather than to present themselves and build relationships. However, while participating in these workspaces, they may inadvertently disclose information about themselves that can undermine their privacy as we will show in Scenario 2 (Section 2.1.2)

### ***2.1.1 Scenario 1: A Social Network Site***

Natalie Blanchard is a member of the biggest social network site in the world: Facebook. She has a profile page, detailing information about her person, her hobbies and preferences. Through Facebook, she stays in touch with a number of contacts, both close friends and acquaintances. Natalie knows that users must be careful about sharing their personal information in social network sites because of privacy issues, so she has changed the privacy settings of her profile to ‘visible to friends only’. This means that only the members of her contact list can see the information she posts there. Natalie regularly posts pictures to her profile page, for instance of a trip she took to the beach, or of parties that she attended with friends.

One day in 2009, Natalie receives a message from her insurance company, telling her that they will terminate the monthly payments she has received for the last year and a half because she is on sick leave – she had been diagnosed with depression in 2007. Inquiries reveal that the insurance company had used Natalie’s Facebook page to investigate the validity of her ongoing claim for monthly payments, and had used the pictures of her, happy at the beach or laughing with friends, to conclude that Natalie was unjustly receiving these payments. It remains unclear how the insurance company gained access to the profile page, if Natalie had indeed shielded it from everyone who was not on her contact list.

This scenario reveals that unintended audiences may sometimes access personal information in social network sites, and thus receive information that was not posted with them in mind. Based on what they see there, these unintended audiences may draw conclusions, and even undertake actions, that may harm the individual involved. Sharing information without having a clear grasp of the makeup and the extent of the audience, as is the case in social network sites, may thus have serious repercussions for users.

### 2.1.2 Scenario 2: A Forum

Hannes Obermaier works as a salesman in a big electronic company and his favourite hobbies are his family and gambling. He plays poker well and has even won a small amount of money in an online poker room. Unfortunately, Hannes forgot to indicate this earning in his tax declaration and therefore he has a problem with his tax office. Seeking for advice in this situation, Hannes finds a forum on the Internet where all kinds of questions related to online gambling are discussed. Hannes hopes to find help and creates a forum post in which he describes his problem. After a few minutes, another forum user has written the first reply to Hannes post saying that he has experienced similar problems and asking about some more details of Hannes' case. During the next few days, Hannes spends a lot of time in the forum. He has gotten to know a lot of the other users and with three of them he really feels like they have been friends for ages. Of course, Hannes has told them not only about his problem with the tax office, but he also shared some funny stories from his everyday life and posted a link to a cute picture of his son from his personal homepage.

One day, a new user who calls herself WendyXY appears in the forum and starts to post insults and allegation about Hannes, not just once but repeatedly. The scary thing is that she seems to know Hannes' name, where he lives and for which company he works. Later, Hannes realises that WendyXY may have found his personal homepage since he had posted the link to the picture of his son. His personal homepage contained Hannes' real name and his residence. Knowing this information, it must have been easy for WendyXY to infer where he works since Hannes has briefly mentioned his job in earlier posts and there is only one big electronics company in his local area.

The story becomes even worse when one of Hannes' major clients finds all the allegations about him on the Internet and cancels an important contract for fear of a negative image.

This scenario may seem artificial, yet a similar case was reported by the German newspaper *Zeit* in 2009 [Bur09]. It illustrates that the sharing of personal data with possibly millions of unknown people on the Internet is a critical point from a privacy perspective and may result in negative consequences, such as bullying, cyberstalking or harassment. However, note that the sharing of personal data with an *intended* audience – in the scenario for example talking about the tax office problem with other online gamblers – is the main reason to use forums or similar collaborative workspaces.

### 2.1.3 General Requirements

In both scenarios, users' privacy could be protected through the use of *audience segregation* [Gof59], i.e., the compartmentalisation of different social circles, so that individuals can show different sides of themselves in each of these circles, without

running the risk that information from other domains of their lives undermines the current self-presentation. In technical terms, one could argue that *selective access control* based on access control rules specified by the user is one feasible means to realise audience segregation. Due to the differences between social network sites and collaborative workspaces such as forums, different requirements emerge with respect to the way selective access control is implemented.

In social network sites, users are directly connected to other members, whom they know (at least to some degree). This means that generating selective access control in social network sites refers mainly to the fact that users must be able to make information visible to specific contacts (yet not others). This can be realised by enabling users to create multiple profile pages in a social network site, and to cluster contacts in subgroups so that information disclosure becomes more targeted and precise. We will outline these principles in more detail in Section 2.2.

In collaborative workspaces, such as forums, users are generally not connected to others, and they may not know any of the other members of the workspace. In these collaborative workspaces, privacy-enhancing selective access control can be realised based on general properties of the intended audience, without having to “know” each user in particular as we will show in Section 2.3.

## 2.2 Two Prototypes for Privacy-Enhanced Social Networking

### 2.2.1 Introduction

Social network sites are one of the most flourishing branches of the social web. In 2008, Facebook, the biggest social network site of all, had over 100 million users [Gri08]. In October 2010, the same social network had grown to more than 500 million *active* users worldwide – ‘active users’ are defined by this network as individuals who access the network every single day on average [Fac]. Users who have accessed the network only once, or use it less frequently, are not even counted in this number anymore, which means the total number of users must far exceed the phenomenal 500 million that Facebook focuses on. And Facebook is not the only social network site. There are literally thousands of different social network sites on the Internet. Roughly 200 of the biggest ones are listed on a Wikipedia page, which reveals that, on average, these sites gather hundreds of thousands of unique users and many go up to millions [Wik]. In March of 2009, Nielsen Online, an information and media company that analyses online audiences and their behaviours, published a report that shows that the use of social network sites and blogs is now “*the fourth most popular online activity, ahead of personal email*” [BM09].

While social network sites have become immensely popular worldwide, at the same time stories of privacy breaches on these sites are a regular topic in popular press. There is an abundance of examples of social network site users who have not managed to find a job [Man10], lost a job [Yok07], had to paid extra income



taxes [Ind08], or lost their right to health insurance payments [New09] because of information they had revealed through social network sites. Many empirical studies have been conducted in the previous years to reveal (a) how users perceive privacy on social network sites, and (b) how (mis)perceptions of privacy in relation to users' own behaviour can lead to privacy violations in these domains.

### 2.2.2 Privacy Issues in Social Network Sites

One of the most fascinating aspects of users' self-presentation in social network sites is the fact that they put such detailed and personal information about themselves in their profiles [Tuf08, YQH09]. In an article on the privacy risks for individuals using Facebook, Grimmelmann points out:

"Facebook knows an immense amount about its users. A fully filled-out Facebook profile contains about 40 pieces of recognizably personal information, including name; birthday; political and religious views; online and offline contact information; sex, sexual preference and relationship status; favorite books, movies, and so on; educational and employment history; and, of course, picture. [...] Facebook then offers multiple tools for users to search out and add potential contacts. [...] By the time you're done, Facebook has a reasonably comprehensive snapshot both of who you are and of who you know" [Gri08].

What's more, "[a]ll of this personal information, plus a user's activities, are stored in essentially a huge database, where it can be analyzed, manipulated, systematized, formalized, classified and aggregated" [RG10]. When viewed from a European legal perspective on privacy, almost all of this information (details on sexuality, religion, politics etc.) is considered 'sensitive' and hence requires "*particular conditions and safeguards [...] when processed*" [EB09].

After an extensive analysis of articles on privacy issues in social network sites, we conclude that such privacy problems may arise in five categories with respect to users<sup>1</sup>. In some respects, these categories overlap with themes we have discussed in the introduction to this chapter, when discussing privacy issues in online worlds in general. However, on social network sites, these themes come together in a particular way. We will discuss each of them in turn.

#### 2.2.2.1 Who is the Audience?

In social network sites, "*audiences are no longer circumscribed by physical space; they can be large, unknown and distant*" [PD03]. It is difficult for users to know who exactly sees their information. The audience, to phrase it differently, is not transparent. On almost all social network sites, users can protect the visibility of the

<sup>1</sup> Privacy issues caused by third party businesses and by the providers of the social network site themselves are another serious threat. These types of privacy issues were discussed in Deliverable 1.2.5 of the PrimeLife project. We have chosen to focus on privacy issues amongst users here only, since those are the issues that we attempted to solve primarily in building our Clique demonstrator.

information they share through so-called ‘privacy-settings’. Generally, these enable them to set the visibility of their profile to one of four options: ‘visible to everyone’ (i.e., all users of the social network site), ‘visible to friends’ (i.e., visible to all the contacts listed in their contact list), ‘visible to friends-of-friends’ (i.e., visible to all the contacts in their contact list *and* to the contacts in the lists of all of these individuals), or ‘visible to no one’. As it turns out, many users – aware of the possible privacy risks in social network sites because of all the media attention for these risks – set their profile visibility to ‘friends-of-friends’. This sounds restricted enough to be ‘safe’ from prying eyes, yet open enough to find new contacts, and to stay in touch with a larger social circle than one’s own strict group. Moreover, many users understand the phrase ‘friends-of-friends’ to refer to, roughly, the group of people one would encounter when going to a friend’s birthday party. However, this is a grave misperception. On average, users in Facebook have 130 friends in their contact list. Aside from the fact that this so-called collection of ‘friends’ must consist of more than real friends, simple multiplication reveals what setting visibility to 130 friends-of friends means: when 130 friends of 130 friends are allowed to view an individual’s profile information, this means that almost 17.000 people have access to that information – a very large group of people indeed. A non-transparent audience may easily lead to privacy problems, because users may unintentionally make information available to the wrong people, or to an unforeseen amount of people.

### 2.2.2.2 Context Collision or Lack of Audience Segregation

Another key element of the fact that users do not have complete control over, or full awareness of, who sees the information they post in a social network site is what Raynes-Goldie has called ‘*context collision*’ [RG10]. When she conducted research on the disclosure of information in Facebook, participants told her they were very frustrated by the fact that in this social network site (and in many others) all contacts are clustered into a single group, without distinction between the myriad of social relations and the various levels of intimacy one has with different contacts in real life. This leads to a

“... flattened Friend hierarchy, where by default, everyone is given access to the same personal information. As a result a user’s teetotaller boss sees the same things as their best friend, the party animal. This can cause problems when trying to decide what to share about yourself, or trying to manage how people from different life contexts might perceive [information]. What is appropriate for a user’s friends to see may not be appropriate for their employer” [RG10].

Context collision entails that individuals are no longer able to meet the various behavioural requirements of the many different social settings in which they normally operate, since one and the same audience sees all of their behaviours. Whereas individuals can keep various social settings separate in real life, for instance because these social settings are connected to distinct physical places (work, home, public space, etc.), in virtual worlds such as social network sites “*intersections of multi-*

*ple physical and virtual spaces, each with potentially differing behavioral requirements*” may arise [PD03].

When phrased in the vocabulary of the twentieth century sociologist Erving Goffman, whose perspective on identity and self-presentation was influential in our analysis of social interaction in social network sites, what users in social network sites lack is a means for ‘*audience segregation*’ [Gof59]. Goffman emphasises the fact that human beings need such a segregation between audiences, since they perform different, possibly conflicting, *roles* throughout their everyday lives, and the impressions they aim to foster in each of these roles must not be contaminated by information from other performances. With segregated audiences for the presentation of specific roles, performers can ‘maintain face’ before each of these audiences. In social network sites, the clustering of social relations into a single list of contacts defeats this important feature of social life in the everyday life. Context collision and context collapse in social network sites are caused by users’ lack of means for audience segregation. When the audience consists of individuals from many different contexts of an individuals’ life, brought together in one group to view all of the individuals’ behaviours in a social network site, then it is clear that this diminishes the individuals’ chances of protecting and maintaining his various ‘faces’. Thus, it may lead to minor or more serious privacy risks.

### 2.2.2.3 Persistence of Information

As we have seen above, the persistence of information in online worlds forms a threat to users’ privacy. As Palen and Dourish say, “*the recordability and subsequent persistence of information, especially that which was once ephemeral, means that audiences can exist not only in the present, but in the future as well—*” [PD03]. This also applies to social network sites. Information posted on one’s social network site profile may be accessed by (known and unknown) individuals in years to come. Moreover, since information can be copied, saved and stored easily and indefinitely, information placed on one’s profile on a social network site at any particular moment may come back to haunt the individual years down the line. This means that the audience is not only unlimited in terms of its size and makeup (in contrast to audiences in the physical world), but also in terms of temporality.

### 2.2.2.4 Peer Surveillance, Snooping and Gossiping

Users of social network sites can use the search functionality of these sites to find others’ profile pages, and may navigate through the profiles of others using the contact lists of individuals they have befriended. Depending on the visibility settings of each profile page, quite a significant amount of information about others may thus be gleaned. Navigating the profile pages of other users in this way possibly invites socially undesirable or even harmful behaviours. For one, gossiping and snooping are facilitated by it. As Hough points out,

“[t]oday, technology enables us to gossip or otherwise exchange information with millions of people instantaneously. [...] Social network sites such as Facebook, reunion.com, and classmates.com enable the resurrection of those embarrassing youthful escapades and awkward photographs we all wish would stay buried. Often, the postings are captured on camera-enabled cellphones without the knowledge of the subjects and uploaded without their consent, leading to instant notoriety, long-lasting embarrassment, and a loss of reserve that may never be recaptured” [Hou09].

Other researchers have pointed out that social network sites are breeding grounds for surveillance between peers. Adam Joinson writes that

“social networking sites like Facebook may [...] serve a surveillance function, allowing users to ‘track the actions, beliefs and interests of the larger groups to which they belong’ [...]. The surveillance and ‘social search’ functions of Facebook may, in part, explain why so many Facebook users leave their privacy settings relatively open [...]. [Social network sites offer users a] ... unique affordance [...] [to] view other people’s social networks and friends [...]. This ability to find out more about one’s acquaintances through their social networks forms another important surveillance function” [Joi08].

Peer surveillance, snooping and nosing around may all lead to privacy issues for the parties subjected to them.

### 2.2.2.5 Who Controls a User’s Information?

On social network sites, users can create a user profile on which they can present themselves. This profile page is the starting point for setting up connections with other users within the same environment. On the profile page, users can choose what information to share about themselves and as we’ve explained above – to some extent – who can view this information (i.e., everyone, friends only etc.). Therefore, in theory at least, users have some control over the image they create of themselves on their profile page. As research has shown, young people especially perceive themselves as having a considerable degree of control over their disclosure of personal information online, and it turns out that they share such information in full awareness of the associated risks, because they have a high degree of confidence in their ability to manage potentially negative outcomes [BK09].

However, the control that users have over their own profile page and personal information in social network sites only goes so far. Other users can add or change information in a user’s personal profile, put pictures or information about him or her on their own or other people’s profiles, and tag pictures to reveal the identities of those portrayed in them. This can have serious consequences: placing a picture of another person online affects the image of that person to the audience viewing it, and hence may have an effect on the (current and future) self-presentations and impression management of that individual.

### 2.2.3 *Clique: An Overview*

In our research, we have developed two demonstrators that may contribute to solving some of the privacy issues in social network sites that we have discussed above. The first demonstrator is a privacy-enhanced social network site called ‘*Clique*’, in which we have turned theoretical notions on audience segregation and context collision into a real-world online practice. The key solutions implemented in *Clique* are:

- providing users with the option to create their own ‘collections’ in which social contacts can be clustered;
- providing users with the option to create multiple ‘faces’ to mimic the practice of audience segregation in real life;
- providing users with fine-grained options to define the accessibility of their postings and content in social network sites, thus mimicking the rich and varied texture of relationships in real life.

*Clique* was built using Elgg Open Source software for developing a social network site.<sup>2</sup> The key ideas developed in *Clique* are:

- mimicking ‘audience segregation’ in a social network by (1) providing users with the option to create their own ‘collections’ in which social contacts can be clustered, and (2) providing them with the possibility to create multiple ‘faces’ to compartmentalise their online social visibility;
- providing users with fine-grained options to define the accessibility of their postings and personal information in *Clique*.

#### 2.2.3.1 Audience Segregation in *Clique*

The first principle we realised in *Clique* to enhance users’ ability to protect their privacy and to increase their options for adequate and realistic self-presentation was a translation of Goffman’s notion of ‘audience segregation’. This is implemented through two mechanisms:

- Users can divide their list of contacts into ‘collections’, clusters of contacts that are socially meaningful, relevant and efficient for them. Each cluster contains one or more contacts; contacts may be listed in as many different collections as the users likes;
- Users can create multiple profile pages for their account. We call these pages ‘faces’. On each profile page, users can show different ‘sides’ of themselves, thereby mimicking audience segregation through physical distantiation (e.g., showing different sides of oneself at home than at work) in real life.<sup>3</sup>

<sup>2</sup> See <http://www.elgg.com>

<sup>3</sup> Note that *Clique* uses a separate social graph for each face to ensure that audiences are indeed kept separate. The social graph collects all of the contacts a user has in relation to this specific profile page, and designates the relationships between them. Collections are a means of assigning access rights to each of the nodes (i.e., contacts) in the social graph.

As we have explained above, on many social network sites, all contacts in a user's network are lumped together into one category. No distinction is made between the different social spheres to which individuals belong in their everyday lives. This means that all contacts are exposed to the same information shared by the individual, regardless of how close they are to that individual. This issue has been solved in Clique through the use of 'collections'. By allowing users to create 'collections' within their network of contacts, they can cluster social relations according to their own preferences, and thereby mimic the actual practice of building and maintaining separate social spheres in real life. We have gone to great lengths to ensure that users themselves are in control of the creation and labeling of collections. After all, they themselves know best what the fabric of their own social lives consists of and how it could be divided into relevant and meaningful categories. In Clique, users can choose (1) how many collections they wish to make (i.e., how granulated they want their audience control to be), and (2) which labels to use for each collection. However, to enhance user-friendliness we also provide them with a set of predefined collections, for instance 'family', 'colleagues', and 'acquaintances'.

Below are some screenshots that show the way in which the creation and management of collections is implemented in Clique. [Figure 2.1](#) shows the way in which users can add a new collection to their profile page. They begin by typing in a name for the collection, in this case 'My favourite colleagues'. Then individual contacts from the user's contact list – that is, individuals that he or she has befriended beforehand – can be added to the collection by clicking through the alphabet and selecting those individuals the user wants to include in this collection. The letters of the alphabet are bold if there is a contact whose user name starts with that letter, and grey if not. After selecting one or more contacts to add to the collection, the user can click 'save' and the collection is added to his profile. [Figure 2.2](#) shows an overview of the collections this user has made and outlines how many contacts are in each collection.

The second way of realising audience segregation revolves around the idea of creating 'faces', so that users can show different 'sides' of themselves to different audiences through the use of multiple profile pages. On most social network sites, users are allowed to create only one profile per person, and hence can create only one context in which all of their information is gathered. However, in real life, individuals move from one context to the next throughout their everyday life – they go to work, visit friends, or spend time at home with their families. To solve the risk of 'context collision' many users currently maintain different profile pages on different social network sites. For instance, they have a work-related page in LinkedIn and a page for family and friends in Facebook. This is a time-consuming and cumbersome solution, since users have to log onto different platforms to manage their profiles and keep track of contacts and their activities in each domain.

To solve this issue, we enable users to create multiple 'faces' in Clique to mimic the various social contexts in which they participate in real life (for instance, a work face, a private face, a face for the tennis club etc.). When a user accesses his profile in Clique, his various faces are visualised with separate tabs. By clicking on the tabs the users can access the page attached to that face. [Figure 2.3](#) shows a screenshot

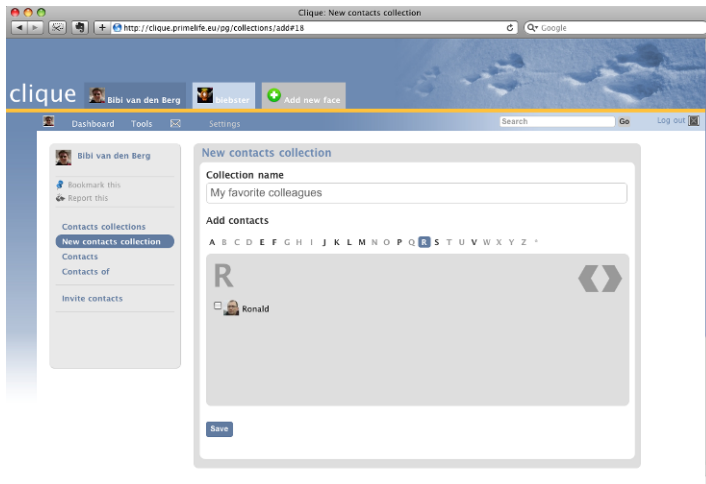


Fig. 2.1: Creating a new collection in Cligue.

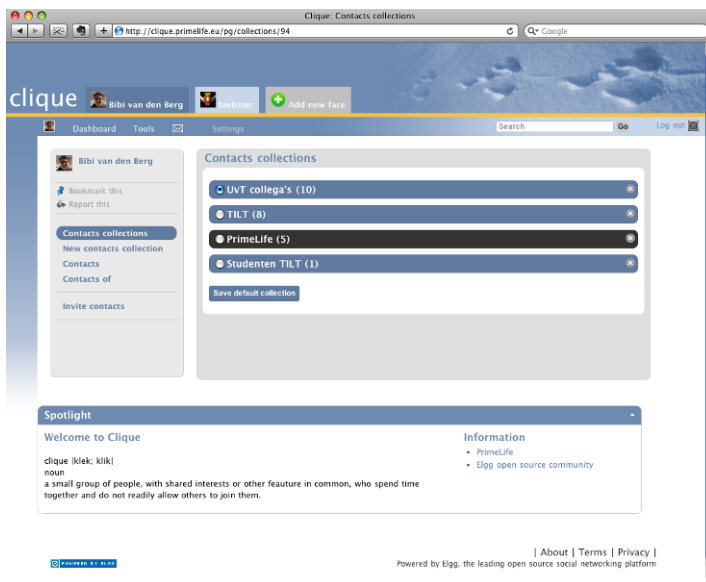


Fig. 2.2: Overview of a user's collections.

of the author's faces in Cligue. New faces can be added by clicking on the tab at the far right, which reads '+ Add a new face'. As the figure shows, existing faces can be enabled, temporarily disabled or removed entirely. Each of these faces has its own list of contacts and its own list of collections. Using tabs to distinguish between different faces is a visually appealing and easy way for the individual to manage his

or her own profile and the various faces contained therein. Information added to one of the tabs is invisible in all other tabs, and hence it is easy for the user to manage who sees what.

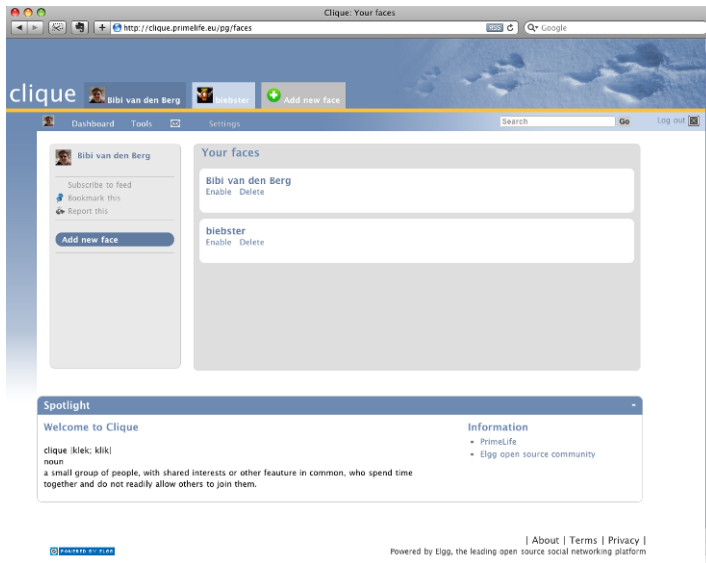


Fig. 2.3: ‘Faces’ in Clique.

### 2.2.3.2 Control Over the Visibility of Individual Items of Information in Clique

To further strengthen users’ capacity to control who sees which information in Clique, we’ve added two more features. First, each time a user posts an item of information (pictures, blog entries, messages), the system asks the user (1) in which ‘face’ he wants to make it available, and (2) to which collections and/or individual users. Second, when disclosing personal information on their profile page, users are also asked, with each separate item of information (i.e., a telephone number, place of residence etc.) to make these choices. These measures further prevent information from spilling over from one context to the next or leaking to unintended audiences. While this feature is demanding on the part of users – it requires them to go through one extra step each time they want to share information with others in Clique –, it is explicitly designed to raise user awareness with respect to audiences and the disclosure of personal information. Moreover, we have designed the interface in a



user-friendly way, so that users can go through this process quite quickly<sup>4</sup> and in an intuitive way. Figure 2.4 shows the screen for setting visibility rights in Clique.

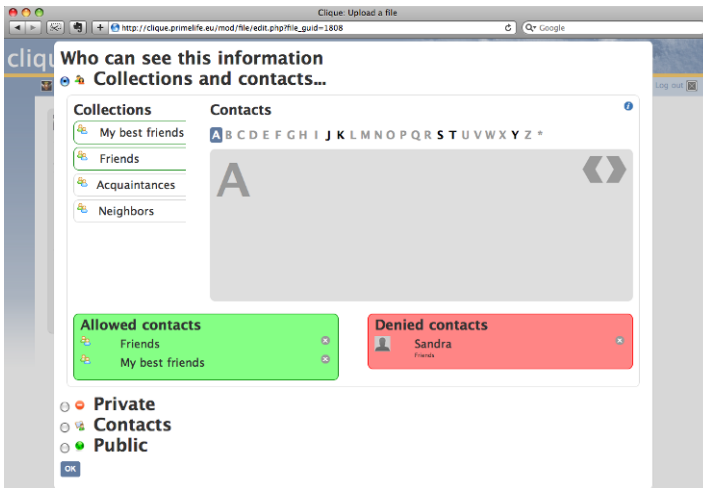


Fig. 2.4: Setting visibility rights to information items in Clique.

### 2.2.4 Scramble!: An Overview

Another demonstrator we have developed in the PrimeLife project to solve some of the issues surrounding privacy and security in self-presentation on social network sites (and also in other Web 2.0 environments), was an encryption tool called Scramble!. In previous years, several researchers have pointed to the need for access control through encryption in social network sites, like Facebook, MySpace or Twitter. Some of their ideas have been turned into tools such as Lockr<sup>5</sup> and FlyByNight [LB08]. However, these tools all have shortcomings.

For instance, Lockr uses trusted third-party storage for the hidden information, which means that the user does not have full control over his own data, but rather has to place trust in a (commercial) third-party to guarantee a safer use of social network sites. The other tool, FlyByNight, only works on Facebook, and relies on the servers of the social network itself for encryption key management. The decryption algorithm is implemented in JavaScript and retrieved from Facebook. Consequently, while FlyByNight is browser-independent and portable, its biggest disadvantage is

<sup>4</sup> If a user decides to display the information to his 'default' collection, it is only one extra mouse click. If the user decides to share it with other collections and/or individuals, this still should not take more than a few seconds to accomplish.

<sup>5</sup> See <http://www.lockr.org/>

that it is not secure against active attacks by the social network provider, Facebook. Both of these tools thus contain serious privacy risks.

Scramble! – the tool we built – provides a solution for the attacker model limitations of Lockr and FlyByNight. It relies primarily on the user side and has no dependencies on any specific social network site, as in the case of FlyByNight. Moreover, it does not use a third-party to store information, as is the case in the Lockr project. Also, what our access control mechanism enables, and what all other existing ones lack, is *selective access control*. We will explain what this means below. Scramble! has the following key goals:

- it enables users on social network sites to formulate which individuals have access to the content and personal information they place in, or attach to, their profile;
- all of the information (both content and personal data) that users place online is encrypted and will only be visible to those contacts and/or collections that have the appropriate access rights. Individuals and collections with no access rights, cannot see the information, and nor can the social network site's provider;
- to aim for ease-of-use in order to strike the difficult balance between usability and privacy for general users.

#### 2.2.4.1 Selective Access Control in Scramble!

With Scramble!, we have aimed to use cryptographic techniques to enforce access control. In this prototype application, we use an OpenPGP<sup>6</sup> standard [CDF<sup>+</sup>07] to keep social network users' data confidential. One nice feature of OpenPGP is that it supports encrypting to multiple recipients using hybrid encryption, by encrypting the content with a random secret, and the secret with all the public keys of the set of users. We assume that each user holds a public and a secret OpenPGP key pair.

For key distribution, we assume that users exchange their public keys when a friendship connection is established using an authenticated offline channel. Users can also make the public key available using the provider or any key server and distribute the fingerprint by the offline channel. In this way, users can verify the authenticity of the public key. Since Scramble! makes use of OpenPGP standard, it can build on the OpenPGP key management infrastructure, retrieving the key from an online key server by name or e-mail mapping.

As an example of the flow, let Alice and Bob be two users on a social network site. Bob accepts Alice as his friend. He then adds Alice's public key to his key ring, thereby including Alice in a circle of trust. Then, Bob can post encrypted messages that can only be accessed by a selective audience chosen from the Bob's circle of trust, which now includes Alice.

To realise selective access control on social network sites based on these principles, we have built a Firefox extension with several features. First, there is the fact

---

<sup>6</sup> OpenPGP is one of the world's most widely used encryption standards. For more information, see <http://www.openpgp.org/>

that access control is generated through the use of an *encryption protocol* (as said, based on OpenPGP), which enables users to trust that their content and personal information is invisible to anyone who does not have the right access key. Note that this includes not only other members of the social network site, but also the social network site provider. Second, the user himself can *define the access rights* handed out to various members of his own social circle, either to individuals or to collections, or to both. The picture below shows two of the windows in which users can define access rights for particular items of content and for personal information.

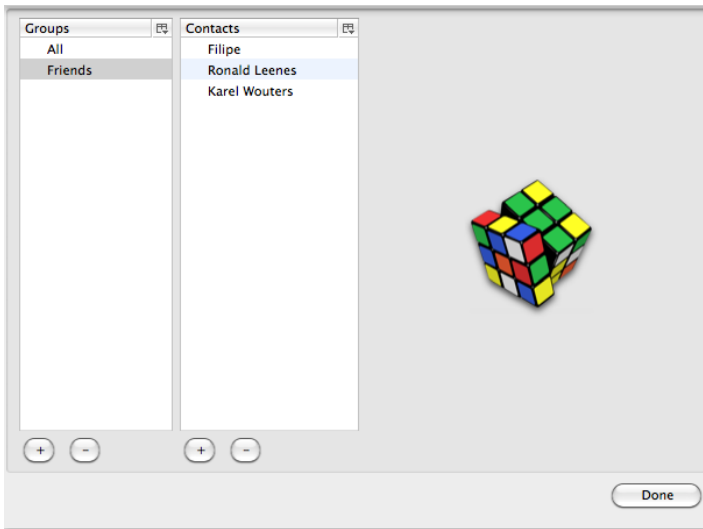


Fig. 2.5: The Scramble! address book, from which selective access rights can be managed.

Since Scramble! is a Firefox extension, it is not linked to a specific social network site, such as Facebook or MySpace, but works across platforms. We have done extensive testing with it in Facebook and MySpace, and also in our own social network site Clique. When a user has the plugin installed and switched on, each time he or she accesses or receives information posted by others, a check is run with regards to whether or not he or she has the proper access rights to read the information. If so, then the content is automatically decrypted and transparently displayed as normal text. Otherwise, if the user has the plugin installed and switched on, but does *not* have access rights to the information, the information is concealed or instead replaced by gobbledygook. [Figure 2.6](#) shows what this looks like in Facebook.

Those who have not *installed* the plugin have no access to the information at all and instead see a so-called ‘tiny URL’ on their screen, a hyperlink referring to an address where the encrypted text is stored instead of the decrypted information

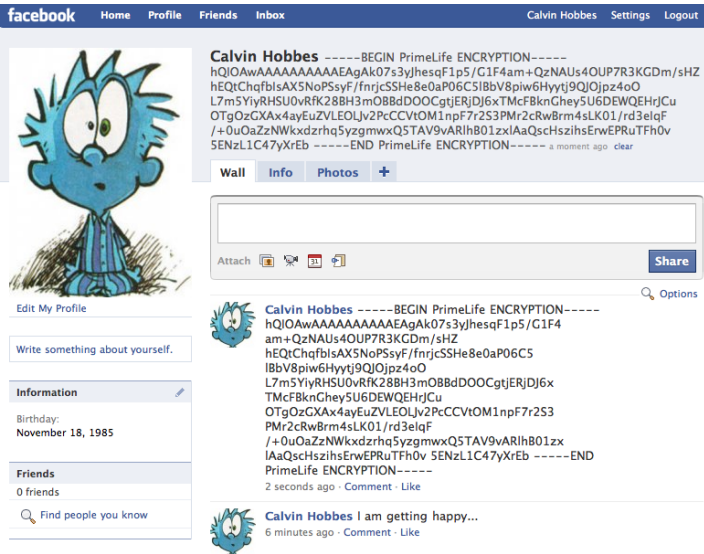


Fig. 2.6: Scramble! is switched on but the user does not have access rights.

placed in the profile by the user. Figure 2.7 shows what this looks like in an earlier version of Clique.

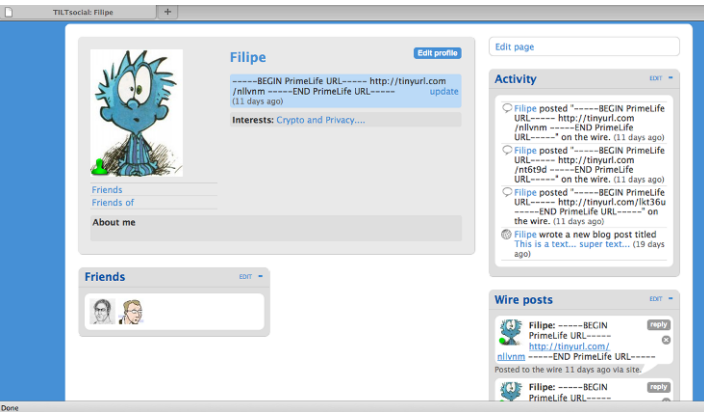


Fig. 2.7: Scramble! is not installed: access is denied and the ‘tiny URL’ is displayed.

#### **2.2.4.2 Scramble! and Web 2.0**

Since this encryption tool is built as a browser extension, it isn't merely independent of the various social network sites, but could also be used in other kinds of Web 2.0 environments, such as blogs, collaborative workspaces such as wikis, and forums. It could be used in any environment in which individuals fill in fields through which they exchange information. Using it is very simple. After installing the plug-in a user can select any kind of text in fill-in fields on Internet pages and simply choose 'Scramble!'; after which the content will only be readable for those who have the right access key. Moreover, because the extension is integrated into the browser the encrypted content will be decrypted automatically for all authorised users, without their intervention. The exchange of keys runs in the background and requires no particular skills from the owner of the information, nor from those gaining access to it. Our extension is simple and aims at striking the difficult balance between usability and privacy for general users.

### **2.3 Privacy-Enhancing Selective Access Control for Forums**

#### **2.3.1 Objectives**

User-generated content in forums may contain personal data in the sense of personal information, personal ideas, thoughts and personal feelings. In contrast to explicit profiles where, e.g., the date of birth is a specified data item saying "12 June 1979," the same personal data can be stated in a forum post which is tagged with the date of writing and says "I got two concert tickets at my 30th birthday yesterday!" In the latter case, it may be not that immediately obvious to the user that she has disclosed her date of birth on the Internet.

The disclosure of personal data in forums and other social software is critical from a privacy perspective, however from a social perspective, the disclosure is necessary since the exchange of information, both personal and non-personal, is the key feature of the application and the primary reason for people to use it. Hence, it is not our objective to prevent disclosure of personal data in forums. We rather want people to be aware of privacy and to enable them to more selectively specify to whom they disclose their data. Access control settings of currently available forums are once specified by the provider and cannot be changed by the particular user. Thus, the user can only decide to disclose information to the groups specified by the providers – in the worst case, this means disclosing to the public – or not to disclose anything at all. Since the first option is not preferable from privacy perspective and the second option is not preferable from social perspective, our objective is to develop a user-centred selective access control system for forums. Forum users should be able to protect their privacy through safeguarding their contextual integrity: data that is disclosed before an intended audience, should not spill over into other contexts and

hence possibly have damaging consequences. That is, a user who wants to share her happiness about her birthday present with some other users (e.g., other people going to the same concert) should be able to specify appropriate access control rules to her post in the forum. Besides the implementation of purely technical means, we further emphasise that it is necessary to sensitise users with respect to privacy in order to get a comprehensive solution. Therefore, we aim at raising awareness of the issue in users as another central goal in the direction of privacy-enhancing selective access control.

2.3.2 Introducing phpBB Forum Software and PRIME Framework

To demonstrate how an existing application can be extended with privacy-enhancing selective access control, we have chosen to build an extension for the popular forum software phpBB [php]. Thereby the main principles of the original system should be preserved. As in other forums, in phpBB, content is always structured in a specific way to make it easily accessible, easy to use, and searchable. In the following, we briefly explain the content structures of phpBB platform, which are illustrated in Figure 2.8.

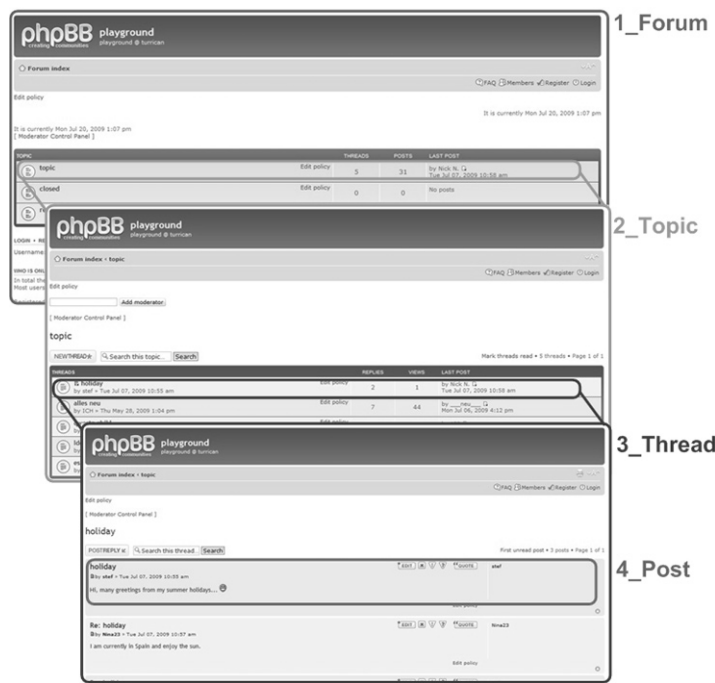


Fig. 2.8: Overview of the hierarchy of a phpBB forum.

The top level resource is the forum itself, which is assigned a title and presented to the user when she first enters the platform. An administrator is responsible for managing all general issues of this forum. The forum is subdivided into topics that each address a different subject matter for discussion. For each topic, moderators are responsible for assuring compliance of the content with ethical quality and forum rules. Thus, they have the possibility to change subjects or content of posts, to lock or even to delete posts. Individual discussions focusing on particular aims are called threads. These are created with the submission of a starting post to which users can reply by submitting replying post.

PhpBB software is available with so-called “copyleft license” and is developed and supported by an open source community. This means, original phpBB source code is available to the public and fairly well documented. With respect to the technical realisation of our selective access control extension, we rely on privacy-enhancing mechanisms that were previously developed in the European project PRIME [PRI]. More precisely, we used *credentials* and *access control policies* from the PRIME framework.

### 2.3.3 Extending phpBB with Selective Access Control

The most common approaches to access control include the access control *list model*, the *role-based* and the *group-based* access control approaches. All three require a central instance that defines lists, roles, or groups based on user names, i.e., identifiers of user accounts (cf. [PBP10]). However, social and collaborative interaction in forums does not necessarily require an association of users by their names. Therefore, privacy-enhancing selective access control in forums requires mechanisms that are not based on the knowledge and existence of names or other particular identity information. Furthermore, it is important that users themselves can decide what they deem to be sensitive or intimate information, rather than what is evaluated as such by lawyers, computer specialists or other third parties [Ada99]. This is why we argue that users themselves need to be given control over the audience to whom they disclose data, and hence access control rules need to be set by the user, being the owner of a resource (e.g., a post), instead of by an administrative party. This implies that access control rules should be possible to specify not only for the whole forum or for topics, but also for threads and particular posts. We need to consider that forum platforms typically provide the roles “administrator” for addressing technical issues and “moderator” for content-related moderation of topics. Our approach should allow for keeping both roles. Hence, we can list the following specific requirements for privacy-enhancing selective access control in a forum whereby these are generalisable to further kinds of social software:

- Other persons, who should or should not be able to access the personal data are not necessarily known by the user.
- These other persons also have an interest to protect their privacy.

- No administrative party, but each user should be able to define and modify access rules to her contributions, i.e., personal information, expression of personal thoughts and feelings.
- User-controlled and privacy-respecting access control can be applied to different levels of content granularity (e.g., forum, topic, thread, post).
- An administrator of the forum should be able to address technical issues of the platform, but should not necessarily have access to content data.
- Moderators should be able to moderate particular topics.
- The owner of a resource is always able to have access on it.

To address these points in our prototype, we let the user define access control policies together with her contribution indicating the properties a reader has to possess and to prove. In order to protect the privacy also of readers, properties are presentable in an anonymous way and not linkable when repeatedly used. Therefore, we relied on the concept of *anonymous credentials* proposed by Chaum in 1985 [Cha85] and technically realised in the Identity Mixer (short: Idemix) system [CL01, CvH02]. The idea of access control based on anonymous credentials and access control policies is not new in general and was demonstrated in selected use cases for user - service provider - scenarios in the project PRIME ([ACK<sup>+</sup>10, HBPP05]). We built on the results of PRIME, transferred the ideas to social software and demonstrated the practical feasibility of maintaining existing concepts of phpBB platform and integrating privacy-enhancing functionality provided by PRIME framework at the same time.

Using anonymous credentials, everyone can prove the possession of one or more properties (e.g., being older than 18, having more than 10 forum posts with a 5 star rating) without revealing the concrete value (e.g., being exactly 56 years old, having exactly 324 posts rated with 5 stars). In the prototype, credentials are also used to prove the possession of a particular role, which may be required by an access control policy. This implies that the process of creating a new resource includes that the originator of that resource receives the corresponding credential (*cred:Owner-Thread-ID* or *cred:Owner-Post-ID*) from the forum platform and stores it on the local device. The roles *administrator* and *moderator* are realised with help of the credential-based access control approach as well, i.e., the according credentials (*cred:Admin-Forum* and *cred:Moderator-Topic-ID*) are issued to the corresponding persons. Together with a new resource, default access control policies are created, which ensure that users who show the administrator credential or moderator credential get the required access granted to fulfill their roles. The owner of a resource possessing the owner credential always has access to that resource and can modify the access control policies to, e.g., allow other users with certain provable properties read access and maybe also write access to the resource.

In general, credentials are offered by particular trustworthy organisations, so-called *credential issuers*. Credential issuers need to be known to the public, so that everybody has a chance to get credentials certifying properties of the user. More details on the technical implementation of the prototype can be found in [Pri10a, Pri10b].



2.3.4 Scenario Revisited

Returning to the forum scenario from Section 2.1.2, the following alternative story illustrates how access control based on credentials and access control policies in a web forum works. Assume Hannes posts a message to the thread “Online Gambling” in a publicly accessible forum. The access control policy of the thread is derived from the parent topic, which is set to be open for reading and writing exclusively for people who have proven to be registered to an online gambling website. Hannes additionally restricts access to his post to allow only gamblers who are registered to their site for 3 months at least.

Table 2.1: Example of an access control policy.

(1) Forum:	[(cred:Admin-Forum) OR (everybody[default])] AND	
(2) Topic:	[(cred:Moderator-GamesCorner) OR (everybody[default])] AND	
(3) Thread:	[(cred:Moderator-GamesCorner) OR (cred:Owner-OnlineGambling) OR (cred:memberOfGamblingSite)] AND	
(4) Post:	[(cred:Moderator-GamesCorner) OR (cred:Owner-PostFromHannes) OR (cred:countMonth-memberOfGamblingSite > 3)]	

Whenever someone requests access to Hannes’ post, the access control policy is evaluated according to the hierarchical order of content elements of the forum (cf. [Table 2.1](#)). In our example, step (1) ensures that authorised users are either an administrator of the forum or – since we have chosen a public forum for the example – any regular user. Step (2) specifies that users are allowed to read the topic “Games Corner” if they are a moderator of this topic or anybody else. The latter applies since the example does not specify any restriction on topic level either. Step (3) ensures that only users who are either moderator of the topic “Games Corner” or who are owner of the thread or who are member of a gambling website get read access to the thread “Online Gambling.” Lastly, step (4) determines that only users who are either moderator of the topic “Games Corner,” owner of the post, or member of a gambling website for at least 3 months can read the post created by Hannes. Note that read access to Hannes’ post is only granted if the whole policy (steps 1 – 4) is evaluated to be “true.” Similar to this example for *read access*, further policies can be defined in order to specify *add*, *edit* or *delete* rights of a resource. All users who add a post to a particular thread have the opportunity to further restrict access to their own contribution.

### 2.3.5 Privacy-Awareness Information

Having described the realisation of the privacy-enhancing selective access control extension so far, in the following we introduce a feature that supports users to be aware of their privacy in the forum. More precisely, we developed a phpBB modification (short: MOD) called “Personal Data” and integrated it into the forum prototype. The idea behind the Personal Data MOD is to provide additional privacy-related information in social software in order to raise users’ privacy awareness, help them to better assess their potential audience and eventually enable them to make informed decisions whether to disclose personal data on the Internet. The perception of privacy in social settings depends on the anonymity or identifiability of the users on the one hand, and on the available audience, i. e., who may read and reuse the disclosed personal data, on the other hand. Considering that privacy is only a secondary task for users, presented privacy-awareness information should be easy and quick to understand and not hinder social interactions and communication as primary tasks in social software.

The Personal Data MOD contains two categories of privacy-related information:

**Audience** Hints about who may access user-generated content, e.g., number and/or properties of potential readers. This partly compensates for missing social and context cues in computer-mediated communication [Dör08] and reminds users especially not to blind out the mass of “silent” forum readers.

**Identifiability** Hints about potentially identifying data that is additionally available, e.g., IP address or location information known to providers. This shows that users are not completely anonymous on the Internet and in particular in phpBB forums, but that there are identifiers available.

In the prototype, the hint about the potential audience is coupled with the setting of the access control policies for read access. If no particular policy is specified for the corresponding forum element and the default policy of the upper-lying content element(s) states “allow everybody,” then the Personal Data MOD indicates “all Internet users” as the potential audience for this post (Figure 2.9). However, if an access control policy is set which restricts the potential audience, the MOD makes users aware of this fact as illustrated in Figure 2.10.

### 2.3.6 User Survey

Besides working on the implementation of the prototype for privacy-enhancing, selective access control, we wanted to evaluate whether our approach meets real forum users’ needs. Therefore we conducted an online survey. The survey was available in German and consisted of two parts: First, participants saw a realistic full-screen screenshot of the phpBB forum prototype with two posts in a discussion thread about leisure-time physical activity as shown in Figure 2.11.



Fig. 2.9: Screenshot prototype: Privacy-awareness information about potential audience if access is not restricted.



Fig. 2.10: Screenshot prototype: Privacy-awareness information about potential audience if access is restricted.

We instructed participants to imagine that they are the author of the first of the two contributions. An orange box on top contained either privacy-awareness hints or an advertisement. In the case that privacy-awareness hints were shown, participants saw either textual information about the potential audience and their individual current location, or numerical information about the exact number of visitors of the forum within the last week and their individual IP address, or a combination of both. All participants of the survey were randomly assigned to one of the four groups. For the second part of the survey, all participants were shown the same online questionnaire. The questionnaire contained questions about knowledge of technical- and privacy-related terms, use of the Internet in general and of forums in particular and questions related to audiences and access control. We also collected demographic data. A link to participate in the survey was distributed via blogs, mailing-lists and forums on the Internet. Due to this setup, we had a non-random sample as a basis for further analysis. After excluding answers from non-users of forums<sup>7</sup> and from participants who had not seriously answered the questionnaire, 313 valid responses remain. In the following, we report selected relevant results based on those 313 participants. More details about methodology, analysis and further results are provided in [PWG10].

First, to test participants' knowledge and awareness of the potential audience, we asked them who actually has access to "their" forum post that they had seen previously. Second, since we were also interested in participants' ideas and requirements regarding access control, we further asked who they would *intend* to have access. Actually, the forum post that we showed to the participants was accessible for all people with access to the Internet, i. e., all registered and unregistered users, forum providers and Internet providers. The fact that the post from the example was completely public could be learnt from the privacy-awareness display with the textual information (shown for G<sub>2</sub> and G<sub>3</sub>) and there was also a visual cue visible for participants of all survey groups indicating that the post can be viewed without being logged in, i. e. it is visible for everybody with Internet access.

<sup>7</sup> Participants who stated in the questionnaire that they have never even read in a forum are considered as non-users.

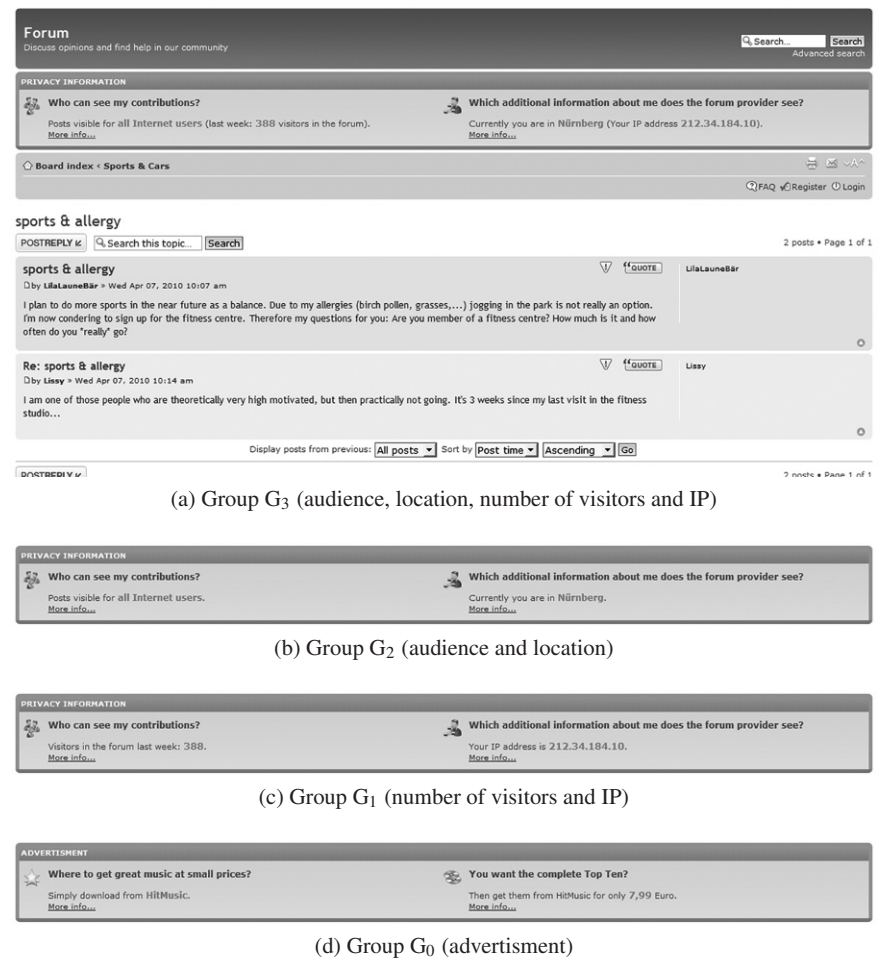


Fig. 2.11: User interface for different survey groups (originally shown in German).

A comparison of the percentages of expected access vs intended access of different audience groups, listed in Table 2.2, reveals that nearly all participants know about and agree with the access to all post for registered members. Also nearly all participants know that the forum provider has access and three-quarters stated that the forum provider should have access. Our results further show that the majority of participants knows that also unregistered visitors can see the post, however only about one-third would *want* unregistered people to view their posts. Hence, there is a considerable difference between the percentage of participants who would let registered users read their posts and those who also would allow unregistered users access to their posts. This finding is interesting for two reasons: First, in most forums on the Internet, anybody can easily become a registered member by providing

Table 2.2: Expected vs intended access to forum posts by different given audience groups.

<i>Audience groups</i>	G <sub>0</sub> n=78	G <sub>1</sub> n=74	G <sub>2</sub> n=86	G <sub>3</sub> n=75	all n=313
All registered users					
expected	96.15 %	97.30 %	95.35 %	97.33 %	96.49 %
intended	89.74 %	100.00 %	96.51 %	96.00 %	95.53 %
Unregistered users					
expected	69.23 %	70.27 %	70.93 %	78.67 %	72.20 %
intended	28.21 %	31.08 %	27.91 %	36.00 %	30.67 %
Forum provider					
expected	98.72 %	95.95 %	95.35 %	94.67 %	96.17 %
intended	66.67 %	75.68 %	75.58 %	70.67 %	72.20 %
Internet provider					
expected	47.44 %	47.30 %	52.33 %	50.67 %	49.52 %
intended	7.69 %	10.81 %	12.79 %	12.00 %	10.86 %

multiple choice questions with given answer categories

a fake e-mail address and choosing a password. This means that practically each Internet user could have access with no great effort, anyway and from this point of view there exists no essential difference between registered and unregistered users. Second, the result indicates that participants want to differentiate who can access their forum posts and that their requirements do not match with current access control settings which are defined by providers or administrators. Looking at the figures for the particular survey groups, we found no statistically significant differences between them (neither with the usual significance level of  $p < 0.05$  nor with  $p < 0.1$ ).

Besides deciding which of the four given audience groups is intended to have access to their forum posts, participants of the survey could specify other groups or share their thoughts about how access control should work in an additional free text field. Indeed, a dozen of the subjects took this opportunity to formulate ideas and said that they would like to authorise particular readers based on special properties or their relationship to them. A selection of comments, which underline real forum users' needs for selective privacy-enhancing access control, is presented below.

*Selection of comments from participants to the question "Who would you intend to access your forum contributions?" (originally posted in German):*

- C1:** "persons I have chosen"
- C2:** "authorised by me"
- C3:** "circle of people that I have defined"
- C4:** "members with at least 10 posts in the forum"
- C5:** "friends"
- C6:** "guests who I have invited"

These requirements can be addressed with the selective access control extension for phpBB forums. The extension enables users to define which properties someone has to possess – by showing either certified or self-issued credentials – for gaining access to users' contributions. It is also conceivable to have credentials that prove an existing relationship, e.g., *being friends in community X*, and using these credentials for specifying access rules. Thereby it is possible to realise relationship-based access control with the selective access control extension without being restricted to it. As argued previously, access control based only on relationships is not suitable for forums in general since this requires that the author of a contribution and the users she wants to give access have to know each other before. This assumption does not hold for web forums, where people with similar interests can meet and discuss without knowing each other in person.

## 2.4 Concluding Remarks

In this chapter, we have investigated privacy issues and issues surrounding the management and expression of identities in social software, with a focus on social networking sites and web forums. The main difference between the two example applications is that in the first case users already have a connection with each other whereas in the latter case potential interaction partners are not necessarily known to each other, e.g., by their user names. We presented appropriate concepts to address privacy-related issues for both types of social software.

First, in order to enhance users' privacy on social networking sites, our solution enables them to cluster their social contacts in their own 'collections' and we provide users with the possibility to create multiple 'faces', i.e., they can maintain multiple profile pages for their account. By defining which profile page should be displayed to the members of a specific collection, users are able to segregate their audiences and actively control who can access their personal data. In case the user does not trust the provider, she can further use *Scramble!* to encrypt all content and share decryption keys only with members of the intended audience. A precondition here is that users can address each other in order to exchange keys.

Second, we have presented the concept and realisation of a web forum prototype for privacy-enhancing selective access control. This solution enables active forum users to specify (a set of) properties which members of their potential audience need to possess. This means that even if the author and the audience are not known to each other, the author controls who can access her contributions to some extent. With *Personal Data MOD*, a feedback mechanism to support users' privacy awareness is integrated in the forum prototype. A user survey complements our work and has shown that the ideas behind the prototype meet real forum users' needs towards user-centred, selective access control.

If the developed selective access control features are used in a very restrictive way, social software users will experience a high level of privacy but a low amount of interactions. Vice versa, if the access control is handled very openly users could

lose much of their privacy. Certainly, it would be inappropriate to stick strict access control policies for every contribution in a forum or to encrypt each single piece of information on a social networking profile. However, having the user-centred selective access control at their disposal may encourage users to discuss issues, which they would not address in public forums, or to state unpopular and uncensored opinions to a specified audience. Neither the privacy-enhanced social networking site nor the web forum with selective access control will completely solve the conflict between sociability and privacy in social software. However, both applications empower users to find their individual balance on a case-by-cases basis.

Considering providers' point of view, it is important to note that privacy is a highly debated topic and providing social software with privacy-enhancing features can be a very good selling argument, especially with respect to people who refused to use social software so far because of privacy concerns.

## **2.5 Acknowledgements**

Many thanks are due to Hagen Wahrig, Richard Wolsch and Joeri de Ruiter for their great work on the practical implementation of the concepts and ideas presented in this chapter.

# Chapter 3

## Trustworthiness of Online Content

Jan Camenisch, Sandra Steinbrecher, Ronald Leenes, Stefanie Pötzsch, Benjamin Kellermann, and Laura Klaming

### 3.1 Introduction

Some decades ago content on which people base important judgment used to be provided by relatively few, institutional sources like Encyclopedias. Since the 1990s the Internet has become an invaluable source of information for a growing number of people. While ten years ago web content has also only been provided by a limited number of institutions or individuals, today's Web 2.0 technologies have enabled nearly every web user to act not only as consumer, but also as producer of content. User contribution is at the core of many services available on the Web and as such, is deeply built into those service architectures. Examples are wikis like Wikipedia, that are entirely based on content contributed by multiple users and modifiable at any time by any of them.

Individuals and organizations increasingly depend on this distributed information, but they face severe trust limitations. In the Web 1.0, it was already difficult to decide to which extent online sources could be trusted. With Web 2.0 the question of trust in online content becomes even more important: Users cannot be sure whether an information is correct, whether the information will be accessible in the future, whether it is legal to use it, and who would assume liability in case the information is incorrect. Users of the Web are not protected against lies and misinformation - think of the recent cases of intentionally false articles in Wikipedia (e.g., BBCNews<sup>1</sup>), or stock price manipulations through misleading newsgroup postings (e.g., CNet<sup>2</sup>).

In fact, with the highly dynamic information flow enabled by the Web, information often takes a life of its own as it can be, for example, published, edited, copied, aggregated, or syndicated; it eventually becomes detached from the context in which it was created and evolves separately. Users do not have cues to determine

---

<sup>1</sup> UK politicians' Wikipedia worries, published Friday, 6 March 2009, accessed 16 Sept. 2010, [http://news.bbc.co.uk/2/hi/uk\\_news/politics/7921985.stm](http://news.bbc.co.uk/2/hi/uk_news/politics/7921985.stm)

<sup>2</sup> Bounty offered for stock tipster, C'Net news, September 5, 1997 12:15 PM PDT, accessed 16 Sept., 2010, <http://news.cnet.com/2100-1023-202979.html>



whether they can trust the information or not. Personal ad-hoc strategies to deal with this, such as trusting only certain websites, are no longer appropriate when the information is dynamically assembled from multiple sources or is passed around, republished, or when the website itself does not perform editorial tasks but entirely relies on its users.

In general ‘Trust’ is a difficult quality to define precisely due to the fact that there are often implicit qualifiers which are determined by the context in which the word is used and that the word is used to mean many different things (even within a single context). Definition is further hindered by the gap between ‘trusted’ and ‘trustworthy’. When speaking about ‘trust in content’ we adopt a limited scope and take this to be *‘the belief that the information is true, accurate or at least as good as possible and the reliance upon this belief’*. When speaking about ‘trustworthiness of content’ we mean that *‘the content satisfies a set of conditions (with implicit qualifiers) defined by another party to justify her well-founded trust in the content’*.

Trust in content is most often derived from trust in a person, entity, or process. As such, there needs to be a binding between content and the person, entity, or process from which trust may be derived. There are two standard approaches to address this binding. The first, more commonly used, consists of trusting the deliverer of the content. These include online news sites, corporate Web sites, and main entries in a blog. The second approach is to include meta-information along with the content that the user may use to assess properties of the content. Such meta-information includes digital signatures on the content itself or the links to external authoritative sources.

The point of ‘trust in content’ is enabling consumers to assess (correctly) the *trustworthiness of content*. Such enabling involves a combination of technical mechanisms, psychological insights, and user education.

Work on technical mechanisms and findings of psychological insights derived by experiments are described within the remainder of this chapter. We first investigate the scenarios of wikis/blogs and derive requirements for technical mechanisms in Section 3.2. As users have both the wish that content can be trusted and the wish of protecting their own privacy a balance needs to be made between both requirements. Based on this we come to technical mechanisms that do not attempt making fully automatic trust decisions on behalf of users, but instead present relevant (primary and secondary) information to them in such a way that they can conveniently and efficiently interpret it as part of the still mental task of arriving at final trust decisions. This act of interpretation can then include whatever additional subjective considerations users wish to apply. Which consideration users apply was studied in experiments we present in Section 3.3. What these experiments basically have shown was that users need education to fully use the possibilities the Internet offers them to establish trust. Finally we present some technical solutions in Section 3.4 that try to aid users in forming their human trust decisions; they do not replace or incapacitate them and should all come along with user education.

## 3.2 Scenarios and requirements

We sketch for the real world scenarios of wikis and blogs how online content is used in different situations and which role the trustworthiness of content plays in these settings. This provides the starting point for exploring a set of mechanisms that allows for the realization of these and similar scenarios. From the mechanisms, a number of more detailed requirements can be derived that were used to build the prototypes documented later on in Section 3.4.

### 3.2.1 Scenarios

There are numerous possible scenarios available on the web that have a need for trusted content. Our choice fell on wikis and blogs as they are already very comprehensive.

#### 3.2.1.1 Blog

A blog is a sequence of generally short articles, often called entries, published on the Web. The entries are produced by an individual or collection of individuals who are the blogs author/s and are often connected by a theme. The entries may consist of text or multimedia, as in a video blog (vlog) or a photographic blog (photoblog). We interpret the term blogs in a relatively broad sense, i.e., not just encompassing individuals online journals, but all content that is ‘pushed’ over RSS or Atom protocols, and other similarly structured content (e.g., from electronic mailing lists) that is easily transformed into the common format. The popularity of blogs as a medium derives from the low cost of production and distribution of content. A direct consequence of this is a large base of content producers and of topics addressed. The issue of whether online information can be considered trustworthy is especially urgent when new information arrives that has to be acted on quickly such as blog articles that convey important news. Often these articles are published by an initially unfamiliar source of origination. Examples are:

**In-company weblog:** Employees of a multinational company (or members of another organization) consume news from similar sources in the form of blogs. Not each individual participant may have the capacity to judge each piece of information in its self-contained form (for a start, it may be phrased in a foreign language), yet the entire ‘crowd’ of members can form an enhanced overall view for ‘inside’ members on augmented ‘outside’ information. This scenario was investigated with a demonstrator outlined in Section 3.4.1.

**(Medical) selfhelp:** Private individuals who consume health-related information (e.g., consider treatment options adjacent to interviews with their physicians), and have obvious warranted interest that this information be trustworthy (e.g.,

‘Is it advertisement? Is it a rumor? What does my insurance company say about it? What is it?’). This scenario points to the importance of including privacy-friendly technology. Experience shows that individuals somewhat differ in their judgments as to the most desirable and practical levels of privacy [Bri98] based on cultural background, politics, age, and other factors; yet privacy is generally held an indisputable right and value when it comes to information that concerns personal health. We use this scenario to investigate which meta-information consumers concentrate on, when making trust decisions on health-related information as outlined in Section 3.3.1.

### 3.2.1.2 Wiki

A wiki is a collection of Web pages, which can be edited online by its users. The objective of a wiki is to support users in collaborative creating and editing of common shared contents. It is possible to link content and to comment on it. Wikis also provide history functionality so that it is easily possible to reset pages to former versions. While some wikis are accessible and editable without authentication, others can have fine-grained access control. A problem of information published in wiki systems is that information can easily be manipulated and be tampered with. An advantage of a wiki system is that information can easily be corrected. The weakness of the system is therefore also its strength, at least if the user base is sufficiently large, committed and knowledgeable. To prevent misuse or vandalism, most wikis try to adopt the strategy of making damage easy to undo rather than attempting to prevent it in the first place. A widely known example for this is the history function with which one can restore pages to older versions. One of the major difficulties in wikis is that it is hard to establish whether information is reliable. The reliability of information depends on the author(s) of the content. Wikis may therefore adopt different mechanisms to control who can create and edit information. One mechanism is that of captchas in conjunction with a text edit field. A captcha is a type of challenge-response test used to ensure that the response is not generated by a computer. Other mechanisms introduce a waiting period before an editor can contribute to a wiki which aims at preventing spur of the moment modifications. The English Wikipedia, for instance, requires new users to wait at least four days before they can contribute. This prevents, or at least delays, rogue automated programs to make edits. Another example is the Portuguese Wikipedia where a user has to make a certain number of edits to prove his trustworthiness and usefulness as an editor. The German version of Wikipedia is currently testing an extension called Flagged Revisions which lets trustworthy authors assign sighted or certified tags. The conditions for an author to be able to assign sighted tags are restrictive in days of having an active account as well as number of edits. The certified tag is work in progress.

We investigate this scenario both with a prototype in Section 3.4.3 and a related experiment described in Section 3.3.2.

### **3.2.2 High-level mechanisms**

By providing consumers with a technological means for not only viewing the primary information online but in the context of related assessments by others whom they are acquainted with, and who in turn may be better acquainted with the primary information, we can facilitate more educated trust decisions that are of benefit to consumers. Trust is ultimately a personal decision. Different individuals may make different choices even when presented with the same ‘objective’ evidence, and not everybody is able or even willing to express what exact considerations go into their respective trust decisions.

#### **3.2.2.1 Core mechanisms**

What technical mechanisms can do on a functional level to assist users is to ensure that a user that consumes meta-data can objectively know that it is related, who authored it (in an absolute or pseudonymous sense), and that it has not been tampered with. We can distinguish the following mechanisms on a functional level:

**Evaluating trustworthiness:** This refers to the process of condensing all available meta-data (such as ratings) that belongs to a piece of information. It forms part of a process that ultimately leads to a binary trust decision on the information whose trustworthiness is under consideration. As an intermediary step, a numeric score for a piece of content may be calculated automatically, which users may then base their final judgement on.

**User reputations and certifications:** The assessment of trusted content depends on who provided a piece of information (or who provided secondary information about it). User can collect so-called certifications and ratings that are aggregated to a reputation. User reputations serve to characterize users in ways that are meaningful to other users when it comes to judging them as sources (e.g., highly reliable source on a scale from 1 to 5).

**Binding metadata to data:** The trust model assumes that when a user does not know whether to trust some piece of information, she can triangulate by taking other, secondary information (meta-data, such as ratings) from other users into account. This entails mechanisms for strong bindings of information to its pursuant meta-data.

#### **3.2.2.2 Supportive means**

Beneath functional mechanisms users need supportive means to deal with the meta-data they got and to provide this meta-data to others:

**Privacy-friendly incentive system:** Providing enough users with sufficient incentives for making their efforts on behalf of other users worthwhile is a known challenge to collaborative schemes such as the intended solution. A privacy-friendly

incentive scheme supports one form on (stimulating) incentives that can help mitigate some related problems.

**Trust policies:** These are artifacts that allow users to declare conditions on meta-data (such as ratings or scores) in order for the information to be regarded trustworthy, or before information can be transformed during its life-cycle.

**Anonymous networks:** Cryptographic schemes that rely on zero-knowledge proofs specify which information protocols must and must not carry such that participants can communicate securely, yet do not inadvertently exchange redundant identifying information. In practice, this must be combined with the use of communication channels that do not expose their identities simply in the form of persistent network addresses or other path information. Anonymous networks serve for this purpose.

### ***3.2.3 Requirements of mechanisms***

In the following we make a first iteration of detailing the core mechanisms described above. From the supportive mechanisms we chose privacy friendly incentive system to do this iteration.

A pre-requisite for all mechanisms are the following requirements:

- **Open API:** The system should offer external interfaces that respect relevant open standards for web and service interfaces, such that it be coupled to existing applications in a straightforward manner. This the case of web applications this can have obvious advantages in terms of potential size of user community, remote access, etc.
- **Efficiency:** The system should employ an efficient representation of its (cryptographic) artifacts, both in terms of their in-memory representation and resulting requirements on surrounding protocols.
- **Scope:** The system should be applicable to a wide range of target applications, e.g., by using a decomposition into (a smaller group of) components that are specific to a (set of) application(s) and (a larger group of) general components that can serve any target application.

#### **3.2.3.1 User reputation and certification**

A user reputation and certification system comprises the following mechanisms:

**A user-rating mechanism:** While our primary focus is content, users also can be ranked (providing them a reputation). This mechanism allows a party or process to specify the atomic rating of an individual or organisation (who/which produced the content).

**A rating aggregation algorithm:** A rating algorithm aggregates individual ratings to one single reputation. It may allow weighing a set of individual object or entity ratings which would require weight factors to be specifiable/specified.

**Certification:** When entities rate content, the relying parties should be able to trust that the ratings are actually provided by the legitimate raters. Certification of the raters can warrant for this property. Certificates are basically digital signatures issued by a third party that is trusted by the user and that verifies that a public key is owned by a particular party.

**Web of Trust:** This is a decentralized concept based on recommendation that is used to establish the authenticity of the binding between a public key and a user, for example, a PGP trust model. A network of trust can be generated using such a model. This can be contrasted with the centralized or hierarchical relationship between certification authorities that exists in X.509.

**A mechanism to propagate ratings:** Ratings are propagated over some kind of rating network. A mechanism which models this network and the message exchanged is needed for rating systems.

**A mechanism to store reputation:** There are different ways to store reputations, e.g., it may be stored decentrally on user side or centrally at a reputation server.

These mechanisms have to meet the following requirements:

**Authentication of parties:** Users want to both demonstrate that they can be trusted and also ensure that the parties they deal with are trustworthy.

**Completeness of reputation:** Users want the aggregated reputation to consider all ratings given. During the storage and propagation of reputation it should not be possible for the entities involved to omit certain ratings.

**Pseudonymity of authors and raters:** Users want to rate and provide web content under a pseudonym to not necessarily allow others to link this rating to their real name. In the real world there are also authors who write under a pseudonym and many services in the Internet also allow the use of pseudonyms instead of real names following EC Directive 95/46 [95/46/EC].

**Anonymity of users:** Users want to evaluate reputation anonymously to prevent others from building personal behavior profiles of their possible interests.

**Persistence of reputation:** The same reputation should be available for all pseudonyms a user uses in a context.

**Self-determination of shown reputation:** If there exist only few authors with the same reputation these authors are easily linkable despite of using different pseudonyms because of the same reputation value. Thus, authors should get the possibility to determine how much of their positive reputation they show. Negative reputation must not be omitted.

**Transparency:** The reputation algorithm must be able to show how an aggregated rating was derived on the basis of individual ratings. The system has to be designed in a way, that the user may check the integrity of single ratings as well as the integrity of the reputation.

### 3.2.3.2 Binding Metadata to Data

Secure metadata support comprises the following mechanisms:

A mechanism for combining a piece of data and its metadata in a secure manner:

This mechanism ensures that content and the meta data remain associated, that is that it is impossible to tamper with the content without this being reflected by the meta-data. This can, for instance be achieved by forming signature of the whole.

A mechanism for checking that metadata belongs to its data: This mechanism allows the relying party to check whether the metadata actually concerns the data to which it is purportedly associated. This could be accomplished by offering the relying party a mechanism for checking the signature of the combined bundle.

A mechanism for reliably referring to single instances of a piece of data:

These mechanisms have to meet the following requirements:

**Integrity:** The system must ensure that the combined bundle of data and its meta-data is safe from unauthorized modification.

**Non-repudiation:** The system must ensure that the effective combination of data and its metadata cannot be denied by a user who created a signature on the bundle. This requirement may conflict with the requirement of authors being able to contribute and rate pseudonymously in the system. The harmonisation of these two requirements requires special attention.

**Normalization:** The system should be able to normalize both data and meta-data to account for the fact that (semantically) equivalent forms may be represented by different byte strings, yet should lead to same signature values.

**Transparency:** The mechanism for reliably referring to single instances of a piece of data should respect existing conventions for data references in general. (This can, e.g., be achieved by forming URLs under a new schema.)

### 3.2.3.3 Evaluating Trustworthiness (or any other property of content)

A trust evaluation system for content comprises the following mechanisms:

A mechanism to request content to be evaluated: This mechanism allows a user to specify that certain content needs to be evaluated in terms of trustworthiness, integrity, validity, relevance, etc. The requester may associate a reward or incentive to the fulfillment of the request. The incentives or rewards may be specified in terms of privacy-friendly incentive points (see supportive measures).

A rating mechanism: This mechanism allows a party or process to specify the atomic rating of particular content (i.e., the content-rating). The rating may be based on the entity-reputation of an individual or organisation who/which produced the content, on certain qualities of the content (content-rating) as assessed by the rater or the rating process (in the case of e.g., text analysis).

An aggregation algorithm: A content rating aggregation algorithm aggregates individual ratings to one single content quality rating. It may allow weighting of

single ratings based on a set of individual content ratings which would require weight factors to be specifiable/specified.

A mechanism to propagate ratings: Ratings are propagated over some kind of rating network. A mechanism which models this network and the message exchanged is needed for rating systems.

A mechanism to store ratings: Most likely the content ratings are stored on the content server.

Similarly to the user reputation and certification system these mechanisms have to meet the following requirements:

Availability of reputation and ratings: As a functional requirement, each user of the rating system should be able to access reputations and ratings to estimate the quality of web content.

Integrity of web content and ratings: Users want web content, ratings and reputation to be preserved from manipulations, both in propagation and in storage.

Accountability of authors and raters: Users want a content's authors and raters to be accountable for the web content they provided respectively rated. This requirement may conflict with the requirement of authors being able to contribute and rate pseudonymously in the system.

Completeness of reputation: (same as in user reputation)

Pseudonymity of raters: (same as in user reputation)

Unlinkability of ratings and web content: Users want to rate and provide different web content without being linkable. Otherwise behavior profiles of pseudonyms (e.g., time and frequency of web site visits, valuation of and interest in specific items) could be built. If the pseudonym can be linked to a real name the profile can be related to this real name as well.

Anonymity of users: (same as in user reputation)

Confidentiality of ratings: Although a reputation system's functional requirement is to collect and provide information about a reputation object, raters might prefer to provide only a subset of their ratings to a specific group of other users while keeping it confidential to all others.

Liveliness: The system may allow existing content ratings to be replaced by novel ratings. This may even be required on the basis of new information, for instance when a rater turns out to have provided unwarranted ratings.

### 3.2.3.4 Privacy-Friendly Incentive System

A suitable privacy-friendly incentive system comprises the following mechanisms:

Obtaining privacy-friendly incentive points for circulation: This mechanism allows users to obtain a collection of privacy-friendly incentive points from a reserve. Points in this collection will effectively enter circulation, and the reserve will enforce an overall policy on the flow of incentive points (e.g., maximum issued number linked to monetary equivalents in users' accounts).



**Exchanging privacy-friendly incentive points:** This allows a party to offer incentive points for certain online transaction and to transfer those points to another party once a transaction has occurred.

**Removing privacy-friendly incentive points from circulation:** This allows parties to return privacy-incentive points to a reserve. Such points will be withdrawn from circulation, and the submitting user will typically receive some suitable form of other compensation (e.g., monetary deposit to her account).

These mechanisms have to meet the following requirements:

**Pseudonymity:** Users must be able to offer and receive privacy-friendly incentives under pseudonyms and without the need to reveal their real identities.

**Double-spending:** The system must be able to detect when users try to cheat by spending the same privacy-friendly incentive points on multiple parallel occasions (i.e., they overspend). Double spending must lead to certain disciplining behavior, such as revealing the users identity to warn against future misuse.

**Accountability:** It must be possible to hold parties accountable the actions taken within the scopes of defined mechanisms. For instance, this must be true with regard to the exchange of pending privacy-friendly incentive points, or with regard to disciplining users because of their alleged double-spending.

**Unlinkability:** The system must ensure that uses of different privacy-friendly incentive points remain unlinkable, as long as they spending them responsibly (i.e., do not overspend).

**Off-line:** The system **SHOULD** support off-line use of privacy-friendly incentive points, i.e., two users can exchange such points without a central party (typically the reserve who issued points in the first place) having to become involved. Especially in an off-line scenario it has to be ensured, that double-spending is not possible.

**Distribution of concerns:** The incentive system should allow parties to store their digital artifacts (e.g., privacy-friendly incentive points) locally, and should not introduce unnecessary assumptions for central storage or other processing at a single location. In case of local storage of the digital artifacts, loss of these artifacts is a concern. Should the system be capable of re-issuing lost credits?

### 3.3 Experiments

In the previous section requirements and mechanisms were sketched that may be used to help internet users assess the trustworthiness of online content. Before we describe in Section 3.4 how these mechanisms can be implemented technically we evaluate additional requirements from practical user experiments.

The first experiment we describe in Section 3.3.1 relates to ‘Binding metadata to data’ (Section 3.2.3.2). We want to know which metadata is useful to function as trust markers. Although, as we mentioned earlier, trust ultimately is a personal

decision, there are of course patterns and some data are more relevant trust makers than other.

The second experiment we describe in Section 3.3.2 relates to ‘User reputation and certification’ (Section 3.2.3.1). Our goal was to find out how private users consider their reputation and other attributes to be to. Based on this we can suggest how to make a trade-off between the metadata other users want to know about content and the trust information others are willing to reveal.

### ***3.3.1 Binding metadata to data***

The first experiment aimed at better understanding the criteria employed by internet users in order to determine which information to trust and which not to trust.

In order to learn more about internet users’ mental trust models and what people consider to be relevant cues regarding content quality and trustworthiness, and how content evaluators handle rating content, we have conducted a few experiments. Questions guiding this research were:

- What are relevant properties to be rated? What are the most salient features of content to call it trustworthy (e.g., validity, accuracy, completeness)? Should the quality be associated to the object-quality score (like author reputation is confined to the domain at hand), or will this be unmanageable by end-users (raters and readers)?
- What are relevant author properties to be rated?
- What binding is required between content and author or rater? Math proofs provided by math professors are likely valued higher than those provided by math students, but this does not say anything about the professor’s reputation regarding fast cars.

#### **3.3.1.1 Findings**

Research on credibility assessment of information found online generally demonstrates that factors pertaining to characteristics of the content, i.e., usefulness and accuracy of the information, factors pertaining to authority, i.e., trustworthiness of the source, as well as factors pertaining to the presentation of the information play key roles in people’s credibility assessments [Met07, Rie02, EK02, FSD<sup>+</sup>03, FC01]. However, there appears to be a discrepancy between indicators people believe they use in order to appraise information they find online and indicators they actually use when assessing the credibility of information [EK02]. Internet users typically indicate that their judgements of website credibility is affected by the identity of the source and scientific references. However, results from observational studies demonstrated that people rarely pay attention to these factors and generally spent little time evaluating the quality of information [EK02, FM00, FSD<sup>+</sup>03]. Instead, it seems that the presentation of information and the design of websites are the most

important determinants of internet users' credibility assessments. On the one hand this finding might be somewhat distressing because people might be easily misled by appealing webdesigns and consequently trust information that may be of little value and low quality. On the other hand, and from a privacy protection point of view, the finding that information about the source has little impact on internet users' credibility assessments implies that information about the identity of the author does not need to be disclosed on websites. If internet users pay little attention to features of the source of the information they read on a website, the absence of this indicator will not interfere with their credibility assessments. Consequently, information lacking author specifications will not be regarded as less credible. This assumption is counterintuitive, especially when keeping in mind that people believe that information about the source is the primary factor influencing their credibility assessments. To explore this assumption in more detail, we have conducted an experiment in which we tested which indicators determine whether or not people find information trustworthy and what role author specifications have in this process.

### **3.3.1.2 Experiment and questionnaire**

The experiment and questionnaire were designed to explore which indicators people view as credible when searching information on a wiki. The test subjects were presented with a mock-up of a medical wiki that had to be used to answer questions about an unfamiliar topic (5-HT or Serotonine). The test subjects could enter search terms in the wiki in order to find information to answer the questions. After entering a search term into the wiki, participants received a list of search results in a random sequence. The list consisted of six search results, each providing a few random words about 5-HT along with information about the author of the text (trust indicators). Each of the results bore one of the following trust markers: (1) the name of the author, (2) the title and name of the author, (3) the occupational title of the author, (4) the title and reputation of the author, (5) a certification of the author, and (6) a reputation measure for the website. Participants could then choose one of the hits and received a text that contained information they could use to answer all questions about 5-HT; the text was slightly different for every hit but included the exact same information. Each text was associated to one indicator, i.e., for each subject the text presented for a given indicator (such as name of the author) was always the same. After having received the search list generated by the wiki, subjects could select a hit from this list and read it and then return to the search results list to choose other hits. In addition, they were free to enter as many new search queries as they wanted. Each time participants returned to the list with search results or entered a new search query, they received four questions concerning the expertise and trustworthiness of the source and three questions referring to the quality of the information using 10-point Likert scales with items: competence, knowledgeability, objectivity, truthfulness, accuracy, usefulness, and comprehensiveness of the author). The procedure therefore was: select hit, read text, answer the 7 information

credibility questions and return to the search results list to repeat the procedure, or answer the 3 questions concerning 5-HT.

After submitting their answers to the 5-HT questions, subjects received a questionnaire about search strategies. The purpose of this questionnaire was twofold. First, one question about reasons for selecting one or more additional hits during the task was integrated into the questionnaire as an additional credibility measure. Second, the questionnaire was designed to generally measure people's strategies for searching information on the internet.

The test subjects in the experiment (256 students at Tilburg University, TU Dresden, National University Ireland Galway, and ETH Zurich, resulting in 172 useable response sets) appear to favour the first search result in the search results list, irrespective of the source of the information. The findings of the experiment demonstrate that internet users' credibility assessments are mainly influenced by the position of information in a search list. The position of a hit in a search list was the most important indicator followed by information about the occupational title of the source. Personal information about the identity of the author was not a particularly relevant indicator for trustworthiness, at least not when compared to position in the search list and occupational title of the author. Personal information about the author, such as his name, became a more important indicator as people selected more than three hits from the search list. Information about the occupation or reputation of the author are more relevant than his or her name. In addition to these indicators, a reputation measure of a website was found to influence people's credibility assessments, whereas a certification such as the one used in the present study (author is a member of the American Medical Association), does not seem to be a valuable indicator for credibility. When looking more closely at what the subjects say about the quality of the information it appears that the test subjects believe that information that is provided along with the occupational title of the author has a higher quality than information that is provided along with a certification of the author regardless of the actual quality of the information. It also appears that position of a hit in a search list generated by a search engine is the most important indicator for its trustworthiness for people's first search, whereas indicators providing information about the source become more important than the position for the subjects' subsequent searches. The main reason for participants to visit more than one search result was to look for confirmation of the information already read on the first entry. While these findings demonstrate that people have a strong tendency to rely on the position of a hit in a search list, they indicated that in general professional writing, message relevance and the absence of typos were the most important indicators for trustworthiness of information they found online. In line with the findings, participants indicated that in general information about the source was of relatively little importance in terms of credibility. Actually, presence of contact information, author identification and author qualifications and credentials were rated as least important indicators for reliability on information found online. Interestingly, only 17.9% of the participants indicated that the position of a search result in the list was very important. When comparing the actual behaviour of the subjects with their beliefs, it becomes clear that people believe the position of a hit in a search list is less impor-

tant for their decision to select a hit than it actually is. This discrepancy between indicators people believe they use in order to appraise online information and indicators they actually use when assessing the credibility of information is in line with previous research findings. However, in contrast to previous findings, participants did not indicate that they found information about the author very important, while this information did affect their actual behaviour, albeit to a lesser extent than the position of a hit. Taken together, the findings demonstrate that our test subjects believe they base their decisions whether to choose a hit and rely on the information they find on the internet on characteristics of the content, while actually, convenience, i.e., the position of a hit in the search engine output, mainly determines their behaviour.

### 3.3.2 *User Reputation and Certification*

According to [Ada99], it is more important that what is deemed sensitive or personal data is based on the perception of the individual rather than if the data can be evaluated by third parties (e.g., lawyers, computer specialists). Considering that individuals often claim to have a great need for privacy but behave differently (cf. *privacy paradox* [Pöt09]), we decided to conduct a study with an experimental part to learn how users actually treat their own reputation value compared to other personal data items. In the following, we briefly outline the set up of the study and report key results. This experiment is also published as an outcome of PrimeLife in [KPS11].

#### 3.3.2.1 Study Design

The web-based study consisted of an experiment and a questionnaire. Invitations to participate in the study were posted in several forums and blogs on the Internet and we also distributed flyers in the university library. All participants who completed the study were offered the chance to win vouchers for an online shop. For the experiment, all participants were asked to rate the same articles from a wiki about books and literature according to three given categories. Before participants actually accessed the wiki articles, they did a short literature quiz. By answering four multiple choice questions about famous writers and books, they received between zero and four points. These points are considered as a subject's *reputation*. Subjects were further asked to indicate name, age and place of residence. When rating the wiki articles subsequently, each participant decides whether her

- name,
- age,
- place of residence and/or
- reputation

should be published together with her rating of a wiki article.

Half of the participants were randomly assigned to the experimental group. The experimental group were shown privacy-awareness information, i.e., information about who can see what data about the user, together with each wiki article. The other half of the subjects belonged to the control group and did not receive privacy-awareness information.

After finishing this first part of the study, all participants filled in the questionnaire. In this questionnaire, we asked about the perceived level of privacy in the wiki, about experience with wikis, ratings systems and the Internet in general. We used questions from the applied privacy concerns and protection scale [Tad10] to investigate general caution, privacy concerns and applied protection measures. Finally, we asked about demographic data and whether subjects had given their real name, place of residence and age at the beginning.

We calculated the *Perceived Privacy Index* (PPX)<sup>3</sup> from participants' answers to the questions about how public, private, anonymous and identifiable they felt in the wiki. Each item was measured on a 0 to 100 % slider scale. The higher the PPX value, the more private a subject felt.

### 3.3.2.2 Results

After excluding complete data sets from a few subjects who admitted not to having seriously participated in the study, 186 valid responses remain and were used for further analysis.

30 % of the subjects agreed to publish their real name together with the rating of a wiki article. The disclosure of their real age was okay for 57 %, real place of residence for 55 % and 63 % agreed to have their reputation value published. This means, for each data item there was a considerable share of subjects who wished to keep this information private. If participants indicated later in the questionnaire that they did not provide true information in one of the first three categories, we treated this data item as not disclosed. Since the reputation value was calculated from answers in the literature quiz, lying was impossible.

Further, we used a linear regression model to calculate how the disclosure of these four data items and a few other factors influenced user's perceived privacy in the wiki. The results are listed in [Table 3.1](#) and reveal that there are only two factors that significantly decreased the perceived privacy: the fact that a user has published her name and the fact that a user has published her reputation value. While it is not surprising that a user feels less private after disclosing her real name, we found that also disclosing their reputation value had a similar effect on perceived privacy. According to the results, the reputation value is deemed an even more sensitive piece of data than age or place of residence. Application-independent measures,

---

<sup>3</sup> The questionnaire contained the question "Please indicate to which extent the following adjectives describe your feelings while using the wiki: 0 % (not at all) – [*adjective*]– 100 % (very much)?" (originally asked in German). The PPX is composed of the adjectives "public" (scale inverted), "private", "anonymous", "identifiable" (scale inverted).

i.e., privacy concerns, general caution and technical protection measures, did not play a significant role for perceived privacy in the wiki.

Table 3.1: Regression model,  $n=186$ .

Perceived Privacy Index <i>PPX</i> (dependent var.)	Est.	Std.er	<i>p</i>
<i>Intercept</i>	288.93	33.47	
<i>Application-specific predictors</i>			
Privacy-awareness information available	4.57	12.01	0.704
Name published	−46.66	14.49	0.002**
Age published	−13.54	16.77	0.420
Place of residence published	−21.65	16.06	0.179
Reputation value published	−39.99	14.04	0.005**
<i>General predictors</i>			
Privacy concerns	−1.35	1.10	0.223
General caution	0.22	1.79	0.902
Technical protection	−0.47	1.47	0.750

sign. levels: \*\*\*  $p < 0.001$ , \*\*  $p < 0.01$ , \*  $p < 0.05$

Altogether, the results of our study underline that a user’s reputation value has to be treated as a personal data item. That means that in a reputation system, users should have the possibility to keep their reputation private, or to disclose only an approximated value.

### 3.4 Demonstrators

In the following we give brief summaries about the demonstrators we built corresponding to the mechanisms described in Section 3.2.2. The demonstrators can be used for either the wiki or blog scenario outlined in Section 3.2.1.

#### 3.4.1 Trustworthy Blogging

Blogs are a representative type of Internet content that is continuously updated by many individuals and organizations. Updates occur by adding new time-stamped articles. For instance, news headlines and reports from news organizations are now commonly available as blogs, and numerous individuals are maintaining what resemble online diaries in the form of blogs. We interpret the term blogs in a relatively broad sense, i.e., not just encompassing individuals online journals, but all content that is “pushed” over RSS or Atom protocols, and other similarly structured con-

tent (e.g., from electronic mailing lists) that is easily transformed into the common format.

The issue of whether online information can be considered trustworthy is especially urgent when new information arrives that has to be acted on quickly. This may well be the case with blog articles that can convey important political, economic, or other news

#### **3.4.1.1 Main Idea**

Information found on blogs is often commented upon by other users. The main idea of the demonstrator is to present such comments to users when they read a blog article. That is, the users not only see the blog article but their browser also presents them comments on this article by other users and information about these other users. Thus, by also reading this secondary information she can better assess the trustworthiness of the blog article. This is of course requires that the Internet is searched for such comments and (reputation) information about the users how provided theses comments is available.

The demonstrator implements this idea on the IBM corporate intranet. As here there exists a central directory that provides information about each employee, the demonstrator focuses on finding and indexing comments and making them available to the users. In particular, the issue on how the deal with the reputation of commenting users is not considered. The demonstrator further offers users the possibility provide their own comments on articles they had read.

#### **3.4.1.2 Demonstrator Architecture and Mechanisms**

The demonstrator consists of a central server collecting and indexing comments and of components that display information to the users. The latter include a firefox-plugin that apart from displaying a web site, e.g., a blog entry, also displays the related meta-information such as comments and identity information about the commenting users. It also offers reader means to provide their own comments on a read blog entry to the demonstrator server.

The central server provides two main functionalities: It is first a service which readers can query for meta-information w.r.t. a piece of information, i.e., a blog article they are reading. The meta-information include comments on a blog the demonstrator has found by crawling the net as well as comments and annotations that readers submit once they have read an article. The second functionality is to collect these meta-information and to maintain in index.

Most of the mechanisms needed for the implementations are rather straightforward and we not describe it here. The main technical challenge was to find a mechanism to bind information (e.g., an article) to its metadata (comments but also all other information about the article such as its author or source). Thereby we can in general not assume that pieces of information (text) have a unique identifier



such as a URL. To solve this, we have introduced the concept of a bound URI (BURI). A BURI is computed from a tuple consisting of information and its meta-data. The computation of a BURI involves *normalizing* the information to give it a unique representation, *versioning* of it, and then *binding* the information and the meta-information together (e.g., by a digital signature of the originator of the meta-information).

### 3.4.1.3 Learnings

User generally find the presentation of the collected meta-information very helpful to assess blog articles. However, the motivation to offer comments themselves seems to be rather low. To address this, we have developed a privacy friendly incentive system that is described in the next section.

We finally note that a central server that provide user with meta-information can potentially track which articles which users read. This could be alleviated by having the users to request this service via an anonymous networks. An alternative method is to use mechanisms that hide the query from the server. Also, if the providers of a blog would mirror the related meta-information, this problem would not occur to start with.

## 3.4.2 Encouraging Comments with Incentives

User-generated content often varies in quality and accuracy. Its trustworthiness as well as its quality can be significantly improved by (expert) reviews and comments. As most scientists know, good reviews are time-consuming, that is, come at a cost. Even though community service out of idealism is a common trait for instance in the Wikipedia community, incentive systems can improve the situation for contributors as well as for the contributed content. They aim at reimbursing the review or revision cost by awards, and at invigorating the review process.

Privacy-friendly incentives complement this fundamental goal with anonymity and privacy protection for all users. Therefore, they enable a double-blind peer review process and nurture fairness, impartiality, and rigor. Authors as well as the reviewers of documents can remain anonymous during the entire review process. Such a blind review process is believed to be essential for high (academic) quality and honest comments, even though it sometimes lacks in reviewer accountability.

Our goal is to establish a cryptographic system that reaches high quality standards, while fulfilling the diverse requirements of the involved parties.

We formalize the incentive system as a collaborative document editing system, in which all revisions, reviews and comments are linked to one initial document  $P_0$ . We consider a document version history  $\mathbb{P} = \{P_0, \dots, P_n\}$  as ordered sequence of revisions, reviews and comments associated with the  $P_0$ , where  $P_n$  denotes the most recent revision or review.

### 3.4.2.1 Principals

There are multiple parties interacting with a document  $P$ . We have a clearing house that hosts all documents and organizes the incentive system, in our case the wiki  $W$  component. The wiki has a community of users and each user  $U$  may act in different and multiple roles:

Reader  $U$ : A reader consumes a document  $P$ . Any reader may offer incentives to other users to improve the quality of a document by a review or a revision.

Author  $V$ : An author contributes an initial version or a revision of a document  $P$ .

Reviewer  $R$ : A reviewer contributes reviews and comments for a document  $P$  in exchange for receiving an incentive.

Editor  $E$ : An editor is a privileged user, who may approve or decline document revisions or reviews by authors and reviewers.

We introduce a bank  $B$  to exchange electronic incentives for real-world goods and awards. Users of wiki  $W$  can withdraw fresh incentive e-coins and deposit spent ones as part of our virtual incentive economy. Even though we allow a system with full anonymity, we require each user to register with a trusted identity issuer  $I$  to infuse accountability in the entire review and incentive process. Each user  $U$  obtains an identity certificate  $\sigma_U$  on its identity  $sk_U$  from issuer  $I$ . Our system works with multiple banks as well as multiple identity issuers, we focus on the single-bank/single-issuer case for simplicity. The identity of an honest user is never revealed by the incentive system, whereas the certified identity enforces separation of duties between authors and reviewers, and prevents double-spending attacks as well as vandalism.

### 3.4.2.2 Concepts

In a privacy-friendly incentive system, many anonymous users interact with a single document  $P$ . Incentives may be given before or after a contribution (revision or review). *Pre-contribution* incentives are offered to users to provide a contribution at all and it is independent from the contribution quality. For instance, a reader  $U$  can offer incentive e-coins for any reviewer  $R$  who is willing to contribute a review. *Post-contribution* incentives are offered after the contribution is made and may be dependent on the quality of the contribution. For instance, users can rate the quality of reviewer's contribution and offer reputation e-coins for his work.

In our model, a reader  $U$  explicitly withdraws incentives from a bank  $B$ . The reader  $U$  offers these *pre-contribution* incentives on the wiki  $W$  for improvements on a document  $P$ . The wiki  $W$  acts as a clearing house and it is responsible for ensuring unlinkability by exchanging the spent incentives of reader  $U$  with bank  $B$  for fresh incentives. Once a reviewer  $R$  decides to contribute a review  $P'$ , he submits the review to the wiki  $W$  for inspection by an editor  $E$ . Once the editor  $E$  approves the review, the reviewer  $R$  can obtain the incentives from the wiki  $W$ . As *post-contribution* incentives extension, the number of obtained incentives can be de-

pendent on the review rating or the reviewer can obtain separate reputation e-coins to build a reputation credential.

### 3.4.2.3 Checks and Balances

The privacy-friendly incentive system provides anonymity to all users and balances this property with strong accountability safe-guards. In a fully anonymous system without such safe-guards, malicious users could attempt to manipulate reviews, sabotage other author's work or publish fabrications without accountability. Well known examples of checks and balances to counter those attacks are the separation of reviewer and author/editor, or the binding of reviews and documents to the contributor's true identity.

To achieve accountability as well as separation of duties between roles, we introduce a cryptographic domain pseudonym  $N_{P,U}$  for each user  $U$  that interacts with a document  $P$ . It is a function of the user's true identity  $sk_U$  and the document  $P$  while hiding  $sk_U$  computationally. Therefore, each entity interacting with document  $P$  has one unique pseudonym, which is independent from entity's role. Pseudonyms  $N_{P,U}$  and  $N_{Q,U}$  created for different documents  $P$  and  $Q$  are unlinkable.

## 3.4.3 *Author reputation system and trust evaluation of content in MediaWiki*

### 3.4.3.1 Architecture

MediaWiki, the software used by Wikipedia, probably is the most used wiki-software. Therefore, the implementation of an author reputation system was done as an extension for this application. In the the following we outline how two of the core mechanisms from Section 3.2.2 can be implemented for the wiki scenario from Section 3.2.1. The requirements and design for this prototype are also published as a result of PrimeLife in [KPS11].

User reputations and certifications: For the credibility of authors, an *author reputation system* assigns *author reputation* to authors. This is done initially by using certifications users got outside the system (e.g., a master degree to show expertise in computer science) and transferring them to a reputation value in the author reputation system. Our reputation system allows to set up different fields of expertise and users can have different pseudonyms and different reputations in these fields. We make use of the identity management system developed by the PRIME project<sup>4</sup> (PRIME) for assisting the user in showing pseudonyms and certifications. For showing reputation PRIME was extended. After registering a user's reputation is influenced by the ratings other users give to the content he

<sup>4</sup> [www.prime-project.eu](http://www.prime-project.eu)

creates within the wiki system. Our reputation system uses the natural numbers (including 0) as set of possible reputation values. As users manage their reputation on their own, one is able to omit single ratings. To avoid, that users omit negative values, our system uses only positive ratings. Therefore it is a disadvantage, to omit any value.

**Evaluating trustworthiness:** A *content rating system* allows readers of content to judge on it's quality and give a *rating* to it. The content rating systems collects all ratings given, aggregates them to a reputation of the content and shows it together with the content to possible future readers. The rating a user gives to the content influences the aggregation algorithm depending on the reputation the rater shows about himself. The problem with wikis is that information changes frequently. The reputation extension is derived from the ReaderFeedback extension for MediaWiki.<sup>5</sup> Using a wiki as implementation platform brought in additional issues like several authors of a content and that there exist different versions of content that do not all get rated. Our content rating system makes use of 5 stars as possible ratings a content might get.

This means that our overall system consists of the following parts:

- the user-side with the PRIME version allowing for reputations installed,
- PRIME certification authorities for issuing credentials/certifications,
- the wiki server with the PrimeLife-ReaderFeedback-extension.

### 3.4.3.2 Functionality

In the following we describe the basic functionality of the system:

#### Fetching Initial Reputation

Authors may start work with an initial reputation. That means, that proofs of competence certified by an authorized institution can be brought in the work with the wiki by using certifications that have been given to the user. This is done by showing anonymous credentials with PRIME to the wiki server. From this, a certain reputation value an author has is calculated by the wiki.

#### Passive Usage

When browsing a wiki page, which has been rated with the help of reputation extension, the user will see the reputation of the content of this page in form of one to five stars.

The reputation shown may not be the reputation calculated for to the latest revision of a page. This is due to the fact, that there may be no ratings given to the latest

---

<sup>5</sup> <http://www.mediawiki.org/wiki/Extension:ReaderFeedback>

revision which is necessary for the calculation of the reputation value. However, if no ratings have been given to the latest revision, the most recent calculatable reputation value will be displayed. If a user wants to know more details of a reputation value the history of single ratings from which the reputation value was calculated can be shown as well (Fig. 3.1).

Revision	Author	141.76.46.77	141.76.46.54
<a href="#">13:47, 14 December 2009</a> (diff)	<a href="#">141.76.46.14</a>	 ★★★★☆	 ★★★★☆
<a href="#">14:20, 9 December 2009</a> (diff)	(many)		★★★★☆
<a href="#">16:56, 4 December 2009</a> (diff)	(many)	★★★★☆	★★★★☆
<a href="#">15:42, 4 December 2009</a> (diff)	<a href="#">141.76.46.16</a>	 ★★★★☆	★★★★☆
<a href="#">15:40, 4 December 2009</a> (diff)	<a href="#">141.76.46.14</a>		
<a href="#">13:41, 20 November 2009</a>	(many)		★★★★☆

Fig. 3.1: Interface showing the Reputation and Rating History.

The tabular representation contains much information in one view. The raters reputation is shown on top of the table below the name or IP address of the rater. The different icons represent the type of reputation which was shown (e.g., the syringes represent a certain reputation in medical field). The stars below the raters are the ratings, which were given to a single revision of the page. If an author indicated his reputation together with submitting an edited page, this reputation is shown beneath the authors name or IP address.

Editing Wiki Pages

When editing a page, a user is asked if he wants to send his reputation value. This reputation value is needed to calculate the reputation of the page afterwards. The higher the reputation value of the author is, the more impact it will have on the reputation value of the page. For showing reputation a user shows a credential. We make use of credentials that allow greater-than-proofs to allow an author to decide about the amount of reputation he reveals depending on his wish for anonymity. e.g., when having a reputation value of 63, an author may prove that he has a value greater than 20 or greater than 50. The higher a reputation value is, the more impact it will have on the reputation value of the page but as the set of authors shrinks when increasing the reputation value, the anonymity-set of the author shrinks as well.

As every user has to decide on this trade-off on his own, a so called “Send Personal Data Dialogue” asks the user for his reputation value and tries to display the trade-off in a graphical way. This dialogue is shown in Fig. 3.2a.

Additionally, the type of the reputation is important for the calculation. While the topic of a page does not change, the author may have several reputation credentials. e.g., a surgeon may edit a page, the content of which is about gynecology. The reputation credential on surgery may have more impact on the gynecology article than a reputation credential dedicated to dentistry. However, having a credential on gynecology would have the most impact. An author may not only show a credential from his concrete reputation type. Within the issuing process he obtains a more general value automatically (e.g., while issuing a credential about gynecology, one obtains a credential about medicine and a general reputation credential as well). When asked for his credential, the user may decide if he shows the specific credential (which has more impact on the page-reputation) or if he uses the more general one (with the benefit, that the anonymity set is higher).

In addition to his reputation value and type, the user may send some identifier. This gives him the possibility to benefit w.r.t. the increase of his reputation value, whenever other raters give a high rating to the page. However, giving an identifier makes the user linkable of course. The decision about sending the identifier is done with a checkbox shown in Fig. 3.2a on the bottom.

The identifier has to be shown again, when the user wants to fetch a reputation-update later. Fig. 3.2b shows this dialogue.

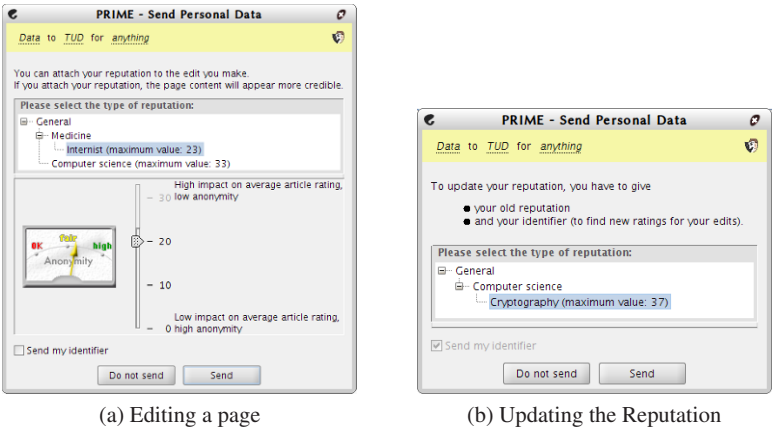


Fig. 3.2: Customized Send Personal Data dialogue.

Rating Pages

In addition to editing, users can actively influence the reputation of a page value by rating it. Therefore, a rating form is shown to the user on the bottom of each

page. With this form a user can give 1-5 stars in four criteria of a page (currently reliability, completeness, neutrality, presentation).

Similar to editing, the user is asked for his reputation value, when he wants to submit his rating. A dialogue similar to the one shown in [Fig. 3.2a](#) is shown to the user when he wants to submit his rating. As also stated in the last section, several properties of the reputation value of the rater influences the impact of the page reputation as well as the anonymity of the rater. Therefore, raters have the same choice authors have between a large anonymity set or a high impact of his rating to the reputation value of the page.

### 3.4.3.3 Lessons learned

The prototype built provides a comprehensive framework for creating author reputation and content evaluation in a wiki by considering the comprehensive and partly contradicting requirements such a system has. This concept could be applied also to other applications or cross applications, especially all applications that are PRIME-enabled. But our first user evaluation has shown that the overall framework is too comprehensive for end users to understand in a first hand and make use of all features, especially the privacy features.

## 3.5 Conclusive Remarks

On the one hand the need for establishing trust in online content is obvious. We could show in our experiments, that the kind of meta information users want to establish trust differs and many users are not aware of which indicators they actually use to establish trust as indicated by their behavior. On the other hand according to our experiments users actively contributing to the the trustworthiness of online content by giving ratings to content have privacy concerns about their user reputation as our second experiment has shown.

We presented a set of functional mechanisms with requirements which help to establish trust in content, namely ‘User reputation and certification,’ ‘Binding Metadata to Data,’ and ‘Evaluating trustworthiness’ which help to establish trust in content. Additionally we gave ‘privacy-friendly incentives’ as supportive means with requirements. These mechanisms were implemented as prototypes for two investigated scenarios, either wikis or blogs.

### **3.6 Acknowledgements**

Many thanks are due to Thomas Gross, Peter Hladky, Christian Hörtnagl, James Riordan, and Hagen Wahrig for their contribution to the design and implementation of the prototypes.





## Chapter 4

# Identity and Privacy Issues Throughout Life

Jaromir Dobias, Marit Hansen, Stefan Köpsell, Maren Raguse, Arnold Roosendaal, Andreas Pfitzmann, Sandra Steinbrecher, Katalin Storf, and Harald Zwingelberg

### 4.1 Challenges and Requirements

Much research and development has been done during the past couple of years to assist users in managing their partial identities in the digital world by several types of identity management [BMH05]. A comprehensive privacy-enhancing identity management system would include the following components [CK01]:

- an Identity Manager (IdM) on the user's side;
- IdM support in applications (e.g., at content providers, web shops, etc.);
- various third-party services (e.g., certification authorities, identity providers).

However, current concepts for identity management systems implicitly focus on the present (including the near future and recent past) only. The sensitivity of many identity attributes and the need to protect them throughout a human being's entire lifespan is currently not dealt with. The digital lifespan is the range of time from the emergence of the first information that is related to the human being until the point in time when no more personal data is generated: from the moment of birth until death. Hence, lifespan refers to the temporary aspects of privacy and identity management and, in particular, to the challenges involved in realising (privacy-related) protection goals over very long periods of time. The area of challenges regarding privacy is vast – even when not considering an entire lifespan (see, e.g., [ENI08]). In the following, we describe which additional problems occur concerning lifelong protection of individuals concerning their privacy in a technology-based society.

#### 4.1.1 Dealing with Dynamics

Our society, as well as the individuals that form them, underly dynamics. We distinguish between *dynamics in the surroundings of the individual* and *dynamics in*

*the individual's ability or willingness of managing her private sphere on her own* as outlined in the following subsections [CHP<sup>+</sup>09].

#### 4.1.1.1 Dynamics in the surroundings of the individual

The dynamics of the effects from the outside world – possibly affecting the individual's private sphere – comprise, among others, technological developments, replacement of administration, changes in law and policies, and – last but not least – the evolvement of society.

The least dynamics we have to deal with is the increasing processing of personal data during one's lifetime. This involves the disclosure of personal data to many data controllers, partially because the disclosure and processing of data is officially required (e.g., because of school attendance, tax liability), partially because the data are needed to fulfil tasks in the areas of e-commerce, leisure, communication etc. Fig. 4.1 shows a simplified model of increasing data disclosure to different data controllers, depicted by coloured stripes. The lighter colours on the right-hand side express that the personal data are not needed anymore for the task to be fulfilled, but the data may still live on in official archives, at Internet services, or in the storage of communication partners [MS09]. The data might not be deleted after the time of death nor after the funeral.

The coloured stripes in Fig. 4.1 might also correspond to several *partial identities* [HPS08], for example (but not exclusively) individuals in different areas of their life. Areas of life are sufficiently distinct domains of social interactions that fulfil a particular purpose (for the data subject) or function (for society). Formal areas of life include education, work, and health care. Informal areas of life cover mainly a user's social network including family, friends, leisure, religion etc. Some of these informal areas might become formal by institutionalisation, e.g., for religion in the form of membership in a church.

Another dynamic results from the development in technology including possible risks. The technological progress of the last decades triggers the transformation of our society towards a computerised social community highly dependent on information. The more structures of our society depend on information, the more important is the role that data plays in our everyday lives. During decades of technological evolution, several methods were invented for storing data in various forms and on various types of media. However, the processes and events in the nature, society, and in the life of the data subject cause failures, which might lead to loss of the data during the lifetime of the data subject. Even if unwanted data loss might not be a common phenomenon encountered within the life of every data subject, it may become an evident and serious problem, which emerges in the lifelong extent of time.

Privacy becomes an increasing problem, e.g., unauthorised access to personal data which enables attackers to read them, link them with other data, or modify them. Personal data, which are assumed to be secure at a specific point in time, may be at risk after some time if no additional precautions have been taken.

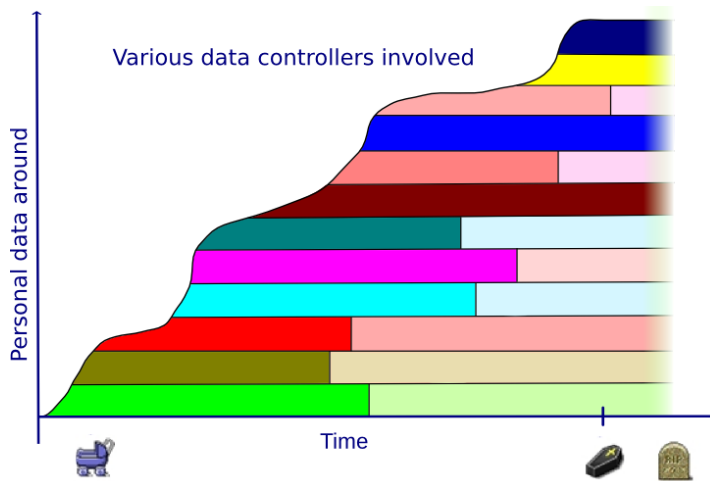


Fig. 4.1: Accumulation of personal data and formation of partial identities.

Also, in the political and societal area, dynamics are to be expected during a period of several decades. In particular, it is not predictable how people in a leading role will interpret today's personal data or which policies will apply.

Lifelong privacy mechanisms need to cover not only the near past and future of an individual, but also need to consider the future prospects for a human's lifetime and also beyond, which means about 90 years. The problem is that we only have experience with computer-based technology for about half of this time, and with the growing exchange of computer data over the Internet, even less than a quarter of this time. This means that all privacy mechanisms (including public-key cryptography invented in 1976 based on cryptographic assumptions) could not be tested in practice for a whole lifetime yet. For the selection of privacy technology (hardware and software), attention should be paid to the following aspects:

- The duration of cryptographic security (based on cryptographic assumptions) should be at least an individual's lifetime. If unconditional security (not relying on cryptographic assumptions, but just on pure probability theory) is possible and feasible, it should be provided.
- Migration of data between different hardware and software needs to be assured. When considering the long-term risks in an unpredictable setting, the sensitivity of personal data is of utmost importance. For the choice and protection level of personal data processed, a categorisation regarding their sensitivity with respect to privacy has to be made according to [HPS08, CHP<sup>+</sup>09].

4.1.1.2 Dynamics in the individual’s ability or willingness of managing her private sphere on her own.

During their lifetime, individuals pass through different stages. A stage of life is defined as follows:

*A stage of life of an individual with respect to handling her privacy is a period in her life in which her ability to manage her private sphere remains between defined boundaries characterising this stage of life.*

The management of one’s private sphere comprises the ability to understand privacy- and security-relevant aspects concerning one’s private sphere, the ability to (re-)act accordingly, and the ability to use appropriate (often ICT-based) means for one’s (re-)actions. Obviously toddlers cannot manage their private sphere on their own, nor can people suffering from pronounced dementia or those in a coma. Even for those who are mentally able to manage their private sphere, it may not be feasible if it requires using technical devices.

Three large stages of life that individuals typically run through are childhood, adulthood and old age, which are depicted in the example shown in Fig. 4.2. It is quite clear that a baby is physically less able than a 10-year-old to interact with technical devices. So the ability of a child to manage her private sphere and her right to be consulted usually increase with her age. However, at least small children are not able to decide on their own how their data are created and processed and how their private sphere can be controlled. Also, adults may have temporary or permanent needs where others support them or even act on their behalf concerning decisions concerning their private sphere. This is true especially in old age. For small children, as well as for very old people, and in the case of emergency, delegation of the right to manage one’s private sphere is needed. For children, these delegates are automatically their parents; in case of emergency or for old people it might be a close relative.

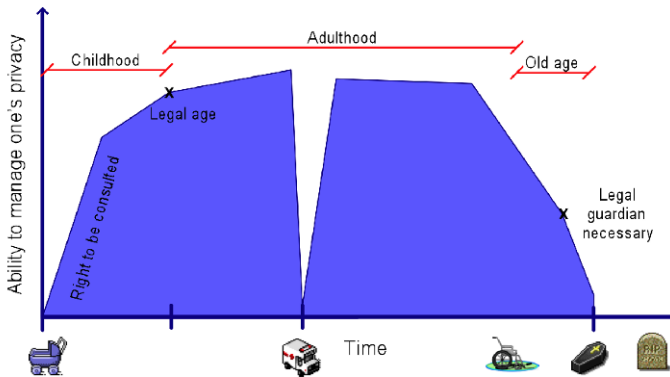


Fig. 4.2: Stages of life: variation in the ability to manage one’s privacy.

Ability in the legal sense would not be partially linear functions as in Fig. 4.2 but more step-like and then remain constant until legal guardianship is needed again.

Sometimes, individuals who in principle are able to manage their privacy on their own want to involve other parties or delegate their privacy control, e.g., for convenience reasons. These individuals may have the ability, but not the willingness to manage their private spheres on their own. Furthermore, both the ability and the willingness might change depending on the circumstances an individual is in and especially depending on what possible data controllers offer to her.

A privacy-enhancing identity management system should support the delegation of duties and authorities. There are three possible situations that might occur regarding delegation from the legal perspective: Firstly, delegation of rights might be made by law automatically for a certain time frame (e.g., for children to their parents). Secondly, delegation might be made willingly by an individual to others for a certain time frame (e.g., delivering mail to others during holidays). Thirdly, delegation of rights of an individual might be initiated by other individuals to achieve delegation of her rights to them or others (e.g., in the case of incapacitating a person), which presumably requires thorough juridical investigation before divesting the person of a right. Delegation can be implemented by different means. Usually, the delegate does not take over the identity of the individual concerned, but receives authorisations to act – often within defined ranges – on behalf or as inheritor, respectively. Technical processes for delegation and digital estate have to be defined in accordance with legal procedures. We come back to this issue in Section 4.1.3.

### ***4.1.2 Digital Footprint***

The difficulties regarding partial identities and identity management in digital environments have another important complicating factor. Next to the partial identities consciously created by individuals to perform actions on the Web, huge amounts of data are collected just because of web activities or other electronic transactions. Every interaction with a communication device, consciously or unconsciously, leads to a data log; a digital trace. By accumulating these data and making connections between the data, extensive digital footprints of individuals can be created.

Digital footprints are data that accumulate in information systems and can be indicated as belonging to one individual. This means that data in this sense is not restricted to personal data only. We outline in the following which data can be sensitive and how it can be linked.

#### **4.1.2.1 Sensitive Attributes**

Some attributes and attribute values usually need more privacy protection than others. According to [HPS08, CHP<sup>+</sup>09], we distinguish the following properties of

identity attributes, which, alone or in combination, pose specific risks to privacy when being disclosed:

- *Data may be static, or changes are quite accurately predictable:* Data which are static over time and are disclosed in different situations enable linkage of related data. Examples for static data are date and place of birth. Similar to static data are those which are quite accurately predictable or guessable because they follow some rules. Examples are data following mathematical rules like the number of children that will only remain or increase. If static identity information is being used for purposes such as authentication, this bears a risk because these data cannot easily be revoked and substituted: For example, the use of fingerprints with biometric access systems.
- *Data may be (initially) determined by others:* Data that the individual concerned cannot determine herself (e.g., the first name) may persist or it may take a significant amount of time or great effort to change them. A special case is the inheritance of properties from others, e.g., the DNA being inherited from the natural parents.
- *Change of data by oneself may be impossible or hard to achieve:* If data are static (see above) or if data are not under the individual's control, wilful changes may not be possible. Examples are data processed in an organisation.
- *Inclusion of non-detachable information:* There are data that cannot be disclosed without simultaneously also disclosing some side information tied to the data. Examples are simple sequence numbers for identity cards, which often reveal gender, birth data and at least a rough timeframe of when the identity card was issued.
- *Singularising:* If data enable the recognition of an individual within a larger group of individuals, the individual may be tracked or located, even if other personal data of the individual are kept private.
- *Prone to discrimination or social sorting:* There are no data that are definitely resistant against possible discrimination forever. This does not need the individual to be identified or singularised. If some people disclose a property and others resist to do so, this already allows for social sorting or positive discrimination.

Note that this list of sensitive properties extends the enumeration of special categories from Art. 8 Data Protection Directive ("personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life"). Because of the sensitivity of the listed personal data, everybody should be careful with related data processing.

#### 4.1.2.2 Linking Data

When taking into account that each and every transaction in an information system leaves a digital trace behind, it becomes clear that the accumulation of data can be enormous and that it thus becomes easier to link data to a single individual. The

more data available, the more unique the combinations of these data are. The more data disclosed to a party over time, the lower the fingerprinting threshold becomes, which means that after a certain amount of time individuals can be uniquely identified and recognised when they appear in a new transaction, even when this new transaction is done from another computer or another location than previous ones [Con09].

The possibilities of linking data that belong to one individual also have their drawbacks on the dynamics in the surroundings of the individual. At first glance, it might seem that changing surroundings can have a positive influence on identity management, because contexts can be separated relatively easily. Nevertheless, linkability appears possible when disclosing information or revealing certain patterns over time, meaning that the different surroundings can be connected as belonging to the same individual as well, therewith creating more challenges with regard to identity management. These challenges are even more supported by the fact that the Internet does not forget. There is no digital oblivion. Once data are revealed somewhere, they persist over time and remain stored in databases, cache memories and so on and so forth.

Obviously, to make the connection between different data sets, some kind of connector is needed. This connector can be created with the help of sophisticated technologies, analysing clicking behaviour, click trails, and key strokes. Linkability between data (sets) can be based on an analysis of the data and revealing patterns. However, this becomes easier when a common identifier exists within different data sets (such as names, registration numbers and IP addresses) and is easiest when this identifier is a so-called unique identifier. Unique identifiers identify a single individual. Unique identification can be supported by governments who issue unique identification numbers to their citizens. These numbers are sometimes even used throughout different domains or areas of life, thereby linking these areas, and the data therein, as belonging to one single individual. In the PrimeLife project, a number of European countries have been investigated on their use of unique identifiers in four formal areas (government, health care, education, employment) and their application range: In the Netherlands a unique identifier, the BSN, is commonly used in several settings which, in principle, allows for the construction of a compound identity, instead of the citizen having distinct identities in different areas. In Germany this is not the case. There is even a separation between the different federal states, which all have their own regime in certain contexts. However, all German citizens of age 16 and above are obliged to have a personal ID card, which is used for identification by public authorities. Up to now, this card only played a role in the physical world. Starting with the 1st of November 2010, the new German eID card will enable trusted services to read out selected attributes over the Internet. Consequently, the holder of the eID card maintains control over which provider is allowed to read out which attributes. Additionally, the new eID card would enable the user to establish a secure pseudonymous communication with a given service provider. Therefore a unique pseudonym will be generated for each relation between an eID card holder and a service provider. France and Austria also have ID cards, but no



unique identifier that is used throughout public services. Belgium, Sweden, Ireland, and Poland do use unique identification numbers, often combined with ID cards.

In the four formal areas, some differences do occur. Some countries have national student cards, others do not. The same goes for electronic health cards. The Netherlands has no electronic health card, but is working on a central system, called EPD, which aims to improve health care by providing access to health records electronically. The system progresses towards a general comprehensive medical identity. In the area of employment, all investigated countries show centralised systems to register people that are unemployed.

### ***4.1.3 Concepts for Delegation***

In the context of identity management throughout life, one focus lies on investigating the necessity of delegation for people who are not able to manage their needs of privacy for a limited time within a stage of life or forever.

Delegation is a process whereby a delegate (also called “proxy”, “mandatory” or “agent”) is authorised to act on behalf of a person concerned via a mandate of authority (or for short: mandate) [HRS<sup>+</sup>10].

The mandate of authority usually defines in particular:

- The scope of authority for the actions of a delegate on behalf of the person concerned and
- when and under which conditions the delegate gets the power of authority to act on behalf of the person concerned.

The delegate shall only act on behalf of the person concerned if the delegate has the actual power of authority and if his action lies within the scope of authority. The simple acting of the delegate with the existence of a mandate while not having the power of authority would not be sufficient. The difference between mandate and power of authority becomes clear in the following example: In working life, the schedule of responsibilities may determine that person A should take over the work of colleague B if the latter is absent. The issuance of the mandate of authority to A is expressed by the schedule of responsibilities, but A’s actual power of authority only comes into existence if B is absent. Otherwise A must not act on behalf of B.

The mandate of authority is issued by the delegator (also called “mandator”). This may be the person concerned herself, but there are also cases where other entities explicitly decide on the delegation (e.g., in the case of incapacitation of a person, the guardianship court rules on delegation) or where the delegation is foreseen in law (e.g., when parents are the default delegates of their young children). The mandate of authority is usually assigned for a specific period of time. Similar to the process of issuing a mandate, changing or revoking the mandate can be done by the delegator, i.e., by the person concerned herself or by other entities. The conditions and processes to issue, change, or revoke a mandate can be defined by the underlying contract or law.

Note that the delegate is not always aware of the mandate of authority or of the fact that he actually has the power of authority. So the delegator should implement an appropriate way of informing the delegate (and the person concerned if she is not the delegator herself) about the mandate and the power of authority.

For supervising purposes of the delegation and related actions by the parties involved, one or more impartial delegation supervisors may be appointed by one or more of the actors. In particular, the person concerned may have the need to check whether the delegate really acts as agreed upon.

The current civil legal framework encompasses several instruments regulating legal representation or agency, which have an effect with regards to the exercise of fundamental rights: For minors, the instrument of parental care is known in civil law. Most of the EC Member States also have legal regulations regarding the representation of children. The Article 29 Data Protection Working Party defined in its Opinion 2/2009 [DPW09] principles regarding children's privacy, which we generalise in the following to the relation of persons concerned and delegates regarding privacy-relevant actions:

- The delegate should act in the best interest of the person concerned. This may comprise protection and care, which are necessary for the well-being of the person concerned.
- Guidelines for delegation should be defined beforehand.
- The person concerned and her delegates may have competing interests. If conflicts cannot be avoided, it should be clarified how to sort them out, possibly with the help of external parties. Note that a delegate does not necessarily stand in for all partial identities of the person concerned, which may lead to additional conflicts of interest of parties involved.
- The degree of delegation should be geared to the capabilities of the person concerned regarding privacy and self-determination. This means that the degree of accountability of the person concerned has to be adapted over time, and regarding privacy-relevant decisions taken by the delegate, the person concerned has a right to be consulted.

Each stage of life has significant question on how to handle identity management and in particular personal data and therefore has different requirements. It is quite clear that a baby is physically less able than a 10-year-old to interact with technical devices. It appears that the privacy protection rights of an individual are exercised by different people during the lifetime. This asks for a delegation system where it is clear for all parties involved who can perform which rights at which moment and in which context. The consequences of the delegate's actions may both influence the privacy of the person concerned and the delegate herself to a certain extent. The following subsections explore various stages of life with respect to delegation.

#### **4.1.3.1 Fruit of the womb**

Privacy throughout life comprises a very early stage of life, the prenatal phase of an individual. Even in this stage of life, there might be the need to protect personal data, for example, considering the privacy implications of prenatal DNA tests. In many EU Member States, there are discussions about the issue of genetic analysis and the threat that using genetic data poses for individual's right of informational self-determination as well as potential discrimination. More detailed regulations regarding requirements for genetic analysis and the use of genetic data could be a solution.

#### **4.1.3.2 Children and teenagers**

Growing autonomy is an important issue in the protection of children's rights, in any area of law. The complexity of situations involving minors is based on the fact that children, despite having full rights, need a representative (delegate) to exercise these rights – including their privacy rights.

Data protection for children starts within the first days after birth and the processing and storage of birth data or medical data within the hospital. The protection of the personal data of children resides more or less in the responsibility of parents or legal guardians as delegates by issued by law. But when a child grows up, other responsible persons for data processing in different areas of life may become involved, such as teachers, doctors or supervisors [HPS08].

The rights of the child, and the exercise of those rights – including that of data protection – should be expressed in a way that recognises both of these aspects of the situation [DPW09]. Until a certain age, children have no way to monitor data processing, simply because they are too young to be involved in certain activities. If their delegates (parents or other representatives) decide, for example, to put the child's pictures on their profile in a social network, it is the delegate who makes the decision about the processing of the children's data and gives the consent to do so on behalf of the child. Normally, putting pictures of another person in a social network profile requires consent of that person, the data subject. In the situation described here, the delegate (e.g. parents) is entitled to express the consent in the name of the child. Such situations may put the delegate in the double role – of data controllers, while publishing their child's personal information open on the Web, and, at the same time, of consent issuers as the child's representatives. This double role may easily lead to conflicts. Parents must take great care not to cross the line of the child's best interest when processing the child's data.

It is necessary for the delegate (e.g. parents or other representatives) to listen carefully to the interests of the child at least beginning from a certain age and consider those interests when making a privacy-relevant decision, as that decision is binding for the child [DPW09]. When the child reaches legal age, it may want to change recent decisions of the delegate. Therefore the child needs to know what decisions about processing of personal data were made by the representatives. Af-

terwards, the child needs to give her explicit consent for the processing of personal data. This may be implemented in certain operations in a way that the operator is reminded that the person is over 18 and now the explicit consent is needed. This is relevant in many circumstances, for example, medical matters, recreational activities of the child, school matters, or agreements made by the delegate before the child's majority.

As children and teenagers are in the process of developing physically and mentally, the rights of the child and the exercise of those rights – including the rights of data protection – should be accomplished in a way that recognises these aspects of the situation. Especially the adaptation of the degree of maturity of children and teenagers is a central aspect that has to be taken into account by their delegates. Children gradually become capable of contributing to decisions made about them. It is natural that the level of comprehension is not the same in the case of a 7-year-old child and a 15-year-old teenager.<sup>1</sup> This, in particular, has to be recognised by the children's representatives. Therefore the children should be consulted more regularly by adults, teachers, care-takers or other delegates about the exercise of their rights, including those related to data protection.

The children's delegate should also think about a way to document privacy-relevant decisions so that the children or young adults can later easily understand what personal data have been disclosed to whom and under which conditions. They may also then choose to actively approach certain data controllers to give or revoke consent concerning data processing or to request access, rectification or erasure of their personal data.

#### **4.1.3.3 Adults lacking privacy management capabilities**

For adults that may have temporary or permanent needs to get support or requiring that others act on their behalf concerning decisions concerning their private sphere, the distinction has to be made between delegation for legally relevant actions and non-legally relevant actions. All legally relevant actions regarding the processing of personal data are based on national legal regulations such as delegation or legal guardianship.

In case of non-legally relevant actions, such as help with a social network or the Internet, in general the person concerned can freely decide what to do. The delegator or person concerned could choose a delegate (for example, a care-taker) to act in the name of the person on the basis of a contract to manage the private sphere. Then the person concerned should clearly define her expectations and needs regarding the representation and the power of disposal.

---

<sup>1</sup> The level of comprehension is defined in different ways. For instance the US-American Children's Online Privacy Protection Act (COPPA, Title XII – Children's online privacy protection, SEC. 1302) defines a child as an individual under the age of 13.

#### 4.1.3.4 Deceased people

In situations where a person has deceased, the instrument of law of succession applies. The European Data Protection Directive 95/46/EC assigns the right of privacy and data protection to “natural persons” (Article 1). Deceased persons are no longer regarded as data subjects. Protection against unregulated processing of data concerning deceased individuals in some European legal frameworks<sup>2</sup> is provided by means of a “post-mortal personality right”. In some situations, the instrument offered by the law of succession might not be sufficient – further regulations are needed.

For instance, some users of social networks want their profile to exist even after death or at least would like to be informed as to how the provider handles the personal data and the profile after death. Here, the action of providers of social networks is required to find mechanisms and concepts for handling profiles after the death of the user. Various mechanisms are conceivable, for example, the user could determine how her profile should be handled after death within the registration process (deletion, blocking, proxy to contact, etc.). Therefore, SNS providers need to define clear measures and concepts to determine the handling of profiles after one’s death. In some situations, even the autonomous action of the SNS provider might be essential for the protection of users. For example, if a SNS user dies and the press accesses the SNS site to copy pictures, contacts, etc. of the dead user, the provider has to balance the protection of the users rights and her competence to, for example, block the profile without the consent of the legal assignee (because this has to happen very quickly).

Meanwhile, new services appear on the market that offer to send out secure messages to friends after the death of the user. Their goal is to give people a safe way to share account passwords, wills and other information. When users book the service against payment of a fee, they are given options for when to send messages or to delete some messages permanently after their death. It is problematic if authentication credentials of the user have to be transferred to the service, which opens the way for misuse because it is not distinguishable for others whether the user or the service acts.

#### 4.1.3.5 Delegation based on explicit decisions of the data subject

The Civil law recognises the instrument of legal representation for cases where the concerned individual is fully in possession of her mental capabilities and decides on her own to transfer the exertion of rights to another person (for example, Articles 172 et seq. of German Civil Code<sup>3</sup>). Various reasons exist why a data subject may wish to transfer the full or partial legal authority of representation to another individual (the mandate of authority). For example, a person may simply be unavailable

<sup>2</sup> Such as Germany: so-called “Mephisto decision” of the German Constitutional Court; BVerfGE 30, 173.

<sup>3</sup> English translation of *Bürgerliches Gesetzbuch* (German Civil Code) is available here [http://bundesrecht.juris.de/englisch\\_bgb/englisch\\_bgb.html](http://bundesrecht.juris.de/englisch_bgb/englisch_bgb.html).

for a longer period of time with no access to information technology, which would allow transmitting and enforcing remote decisions (for example, during a scientific or recreational journey to a secluded region). Or a data subject may feel that certain services which are handled online are better understood by friends or even a professional data custodian. Actions of and decisions by the delegate (authorised representative) may have consequences for the fundamental rights of the delegator. The delegator may have, at first glance, authorised the delegate to act on behalf via a mandate of authority, which, for example, only granted authority to the delegate to close one contract on the delegator's behalf. Delivering the contractual duties, however, will possibly also require the processing of personal data. The legal authority to represent a person concerned in closing a contract does include the implied authority to initiate the data processing steps necessary to fulfill the primary goal. The instrument of legal representation based on the data subject's declared intention may also have an effect after the data subject's death. The data subject may during her lifetime lay down a last will which binds the heirs. This last will may also comprise decisions regarding how to treat documents or electronic files containing personal data.

The Art. 29 Working Party defined in its Option 2/2009 [DPW09] principles regarding exercising the right of children. These principles may also be helpful for determining principles on delegation in general, because delegators (persons concerned) may have the problem that delegation in privacy-relevant situations might be interpreted in different ways. This means that one may have different needs on good practice of handling privacy.

## 4.2 Demonstrator

A scenario that is relevant to all areas and stages of life and deals with dynamics and possible delegation - this means the challenges we described in the previous Sections 4.1.1 and 4.1.3 - is the area of backup and synchronisation tools and applications.

Many backup systems and backup strategies, which have been available for many years, are already dealing with the problem of unwanted data loss. However, they are mostly protecting the raw data only and do not involve the data subject, his specific characteristics, social relations and interactions as a part of their scope. Existing backup systems and backup strategies also do not reflect the process of evolution of the data subject during his lifetime with respect to the possible different states he might pass through during his lifetime and which might have an immense influence on his ability to manage his data on his own behalf (e.g., illness, hospitalisation, or death). Additionally, existing systems and strategies dealing with the problem of unwanted data loss do not cope with boundaries among distinct areas of the data subject's social interactions. However, these aspects are nowadays becoming more and more sensible on the level of the data, hand in hand with the massive expansion of the technology.

Therefore, we decided to analyse the problem of unwanted data loss from the perspective of lifelong privacy. We found that current solutions do not provide a sufficient level of data protection when it comes to lifelong extent of time and privacy of the data subject holding the data. Based on our findings, we decided to demonstrate that it is possible to cope with problems amplified by the requirements on lifelong privacy when protecting the data subject against unwanted data loss.

The proposed privacy-enhanced backup and synchronisation demonstrator focuses on the following problems closely linked together under the light of lifelong privacy:

1. Protection of the data subject against unwanted data loss during his lifetime by redundancy and physical distribution of the data;

Our findings resulted in the conclusion that the problem of unwanted data loss can be solved by redundancy and the physical distribution of multiple copies of the data from the lifelong perspective. As far as backup and synchronisation tools are also dealing with the problem of unwanted data loss, we decided to establish the main conceptual pillars of our demonstrator on the backup and synchronisation functionality. In the demonstrator, we are proposing to solve the problem of unwanted data loss by taking advantages of services provided by on-line storage providers which are nowadays available on the Internet (for example Dropbox, Apple MobileMe, Windows Live SkyDrive, Ubuntu One and others) and store multiple copies of the data in a distributed environment. Distribution of potentially sensitive backup data in such kind of environment, however, leads to confidentiality problems.

2. Assurance of lifelong confidentiality of the data subject's data stored in a distributed environment;

The problem of data confidentiality in a distributed and untrusted environment can be solved by the encryption of the data. Encryption must assure that only the authorised Data Subject (whom the data belongs to) is able to operate with his data stored in distributed backups by default and nobody else should have implicitly access to it even after the death of the Data Subject. On the other hand, during the lifetime of the Data Subject, unpredictable situations might occur, which might temporarily or permanently limit him in his ability to access his own data (for instance in case of his illness, hospitalisation or death). This might lead to situations that his data, which might be important for other parties relying on it (possibly in a legal relationship with the Data Subject), is not accessible by these parties when needed (for example important work documents) or is permanently lost.

3. Delegation of access rights to the data subject's backup data allowing other parties to operate with his data if specific conditions are fulfilled; delegation capability of the demonstrator allows other parties authorised by the data subject (whom the data belongs to) to access his backup data in case particular conditions specified by the data subject are satisfied. Delegation of access rights of the data subject's backup data could in general lead to situations where authorised parties with corresponding access rights are not only able to access the desired data but also other data possibly covering other areas of the data subject's life,



which they are not authorised to access. This might, however, not be desired by the data subject himself.

4. Distribution of the backup data according to different areas of life of the data subject and his different partial identities.

Distribution of the backup data according to particular areas of the data subject's life or his different partial identities enables the data subject to manage his privacy in such a way that allows him to physically and logically separate his data related to distinct domains of his social interaction.

Besides the above mentioned problems, additional non-trivial issues must be addressed, which are covered by the high-level requirements on prototypes developed within the PrimeLife project. As far as the demonstrator is based on the backup and synchronisation functionality, it also has to address further privacy-related issues amplified by the backup and synchronisation nature. Due to space limitations, we cannot elaborate all the requirements and possible solutions here. The interested reader is referred to PrimeLife document "Towards a Privacy-Enhanced Backup and Synchronisation Demonstrator Respecting Lifetime Aspects" [Pri10c].

In order to correctly understand the descriptions in the following sections, it is helpful to be familiar with the following terminology:

#### Terms:

**Primary item:** an original item for which one or more backup items are created during the back up action. In a general sense, a primary item can be referred to as any determinate set of data, which has one or more copies called backup items dedicated for backup purposes. A primary item can be a file but it can also be a more specific type of data such as, for instance, an e-mail, a contact, or even settings on the TV.

**Backup item:** a copy of a primary item stored in the backup. A backup item reflects the data of a primary item at the time when the backup item is created. Note that even if each backup item must belong to one and only one primary item, this primary item may not exist during the entire lifetime of the backup item. A backup item can exist in several versions at a particular point of time.

**Backup:** a non-empty set of backup items.

**Backup task:** describes which specific set of primary items should be backed up to which storage provider according to which schedule. The output of a run of a given backup task is a backup.

#### Actors:

**Primary user:** data subject who owns/holds primary items.

**Storage provider:** provides storage space for backups.

**Delegate:** an entity that receives particular rights on the backup from a delegator.

**Delegator:** an entity that has the privilege to delegate rights to delegates concerning a particular backup. In most applications of this demonstrator, the primary user acts as the delegator.



**Delegate candidate:** an entity that was selected by delegator to act as a delegate but does not possess particular rights yet.

**Delegation request:** a request sent to the delegate candidate asking him whether he accepts particular rights from the delegator.

**Credential issuer:** an entity that issues a credential verifying a certain status of the primary user. This status can for example be: “primary user is ill,” “primary user is hospitalised,” “primary user is dead,” or others.

### 4.2.1 Overview of the Backup Demonstrator Architecture

After describing the overall goals and visions with respect to the backup scenario in the previous section, we will report on the current state of the design and implementation of the actual backup demonstrator in the rest of this chapter.

The backup demonstrator consists of three main components:

1. **the core**, which offers the main functionality. The core is written in Java and runs as a background process on the machine that contains the data (primary items) that should be backed up. The core makes its functionality accessible using a REST-like interface.
2. **a Web-based user interface** (called “backup console”), which is written using HTML, CSS, JavaScript and Ajax. It can be displayed using an ordinary web browser. It utilises the REST calls offered by the core to accomplish the tasks desired by the user.
3. **a tray icon** shown in the notification area of the taskbar found in many modern operating systems. This tray icon informs the user about important information and status changes related to the backup process. Moreover, it allows the user to launch the backup console.

#### 4.2.1.1 Basic building blocks used by the core

The core is the central place that provides all the methods necessary to create backup tasks and actual backups, to restore them, to manage delegations etc. Moreover, it manages all the data (configuration information, credentials etc.) related to this.

The entire functionality is provided through REST-like calls. Thereby HTTP is used as the underlying application level protocol. Therefore, the core contains an embedded web server (namely Jetty<sup>4</sup>). The binding between the HTTP URLs and the Java methods is done with the help of the “Java API for RESTful Web Service” (JAX-RS<sup>5</sup>). JAX-RS uses Java annotations, which simplifies the process of

---

<sup>4</sup> <http://eclipse.org/jetty/>

<sup>5</sup> <http://jcp.org/en/jsr/detail?id=311>

making a Java method available as web service. Particularly the Jersey<sup>6</sup> reference implementation of JAX-RS is used.

In case the URL itself does not encode all parameters and data necessary for a given REST call, the HTTP body contains the additional data needed as input for the given REST call. The data transmitted in the HTTP body can be encoded either using XML or JSON. The desired encoding type is set by the client. The marshalling/unmarshalling is done using JAXB<sup>7</sup> (in case of XML encoding) and Jackson<sup>8</sup> (in case of JSON encoding). Both APIs allow an automatic mapping between the internal representation and the external one, thus avoiding any need for manually implementing serialisation methods for every data type used.

#### 4.2.1.2 File system and external storage providers

When it comes to the question of where to store the backups, the backup demonstrator follows the trend to store data “online,” e.g., by using dedicated online storage providers or more generally spoken “in the cloud.” Although storing backup data more “locally” is supported by the backup demonstrator (e.g., on an external hard drive connected to the machine, where data should be backed up), using online storage is the more important use case. The reason for this is clearly not the “following trends” aspect. It is rather driven by the “lifelong” aspects which should be demonstrated.

On the one hand, the backup scenario implies that the data (backups) are available for the entire life of the primary user. Clearly managing a large set of external hard drives would lead to much more burden on the user compared to managing contracts with online storage providers. Moreover, by using online storage, the data is accessible from every place in the world where an internet connection is available. This makes the whole process of delegating access rights to a backup much more feasible. Finally, it is much easier to store the backups as truly redundant copies, which in turn is a precondition for avoiding long term data losses.

On the other hand, using online storage leads to interesting research questions with respect to the “lifelong privacy” aspects. First of all, the backed up data needs to be stored in a way so that only authorised entities can gain access to the actual backup content. Besides this need to achieve confidentiality of the backed up data, the “privacy” of the involved entities (e.g., primary user, delegates etc.) should be assured as well. In our concept, it means that each online storage provider should learn as little information as possible about the involved parties (e.g., who accesses which backup, at which time, how often etc.).

The demonstrator uses the “Apache Commons Virtual File System”<sup>9</sup> library, which allows access to various different local and remote file systems by a sin-

---

<sup>6</sup> <https://jersey.dev.java.net/>

<sup>7</sup> <http://jcp.org/en/jsr/detail?id=222>

<sup>8</sup> <http://jackson.codehaus.org/>

<sup>9</sup> <http://commons.apache.org/vfs/>

gle API. Besides the online storage providers, which are supported by default (like SFTP, WebDAV, FTP etc.), plug-ins for well known online storage providers (like Dropbox or Ubuntu One) were developed. Although these plug-ins were deployed together with the software of the demonstrator, conceptually they could be downloaded from the Internet as well. The general idea is that either some kind of directory service lists available online storage providers and provides the necessary plug-ins or that at least a given online storage provider offers a plug-in for his service on his web site.

A storage provider plug-in would not only implement all the necessary functionality to actually access the online storage but will also provide user interface components which allow a user to create a new account with this storage provider. This in turn comprises e.g., the necessary SLA and the payment.

With respect to the trustworthiness (or more general the properties) of an online storage provider, the main assumptions are related to availability, i.e., it is assumed that some data stored at a given storage provider would be (with high probability) available according to the negotiated SLA. Beyond that, each storage provider is seen as a potential attacker (in the sense of the concepts of multi-lateral security). Especially, it should not be necessary to trust the storage provider with respect to confidentiality or integrity of the stored data. Nor should the storage provider be trusted with respect to the privacy of a given user. Therefore the backup demonstrator needs to implement appropriated measures to achieve these protection goals (e.g., by means of cryptographic mechanisms like encryption, integrity protection etc.).

In order to reduce the linkability between different transactions done by the same user with a given online storage provider (or multiple storage providers), it is assumed that a communication layer anonymisation service is used. If the access to the online storage is based on HTTP (like WebDAV, Dropbox etc.), existing anonymisation services like AN.ON<sup>10</sup> or Tor<sup>11</sup> could be used.

Nevertheless, there usually remains the linkability introduced at the application layer. Because we want to support existing (legacy) online storage providers, we cannot assume that they base their access control on unlinkable anonymous credentials. Rather, the common combination of login/password would be used. In this case, the only way to avoid linkability, e.g., that two backups belong to the same user, a user has to create multiple accounts (ideally using multiple online storage providers). Note that for the sole purpose of demonstration, we plan to develop a special online storage provider that in fact uses anonymous credentials for access control. More concrete, the implementation of this storage provider will be based on the Identity Mixer<sup>12</sup> anonymous credentials, which a part of the PRIME Core<sup>13</sup> developed within the EU FP6 integrated project “Prime”<sup>14</sup> and the PrimeLife project.

---

<sup>10</sup> <http://anon.inf.tu-dresden.de/>

<sup>11</sup> <https://www.torproject.org/>

<sup>12</sup> <http://www.primelife.eu/results/opensource/55-identity-mixer/>

<sup>13</sup> <http://www.primelife.eu/results/opensource/73-prime-core/>

<sup>14</sup> <https://www.prime-project.eu/>

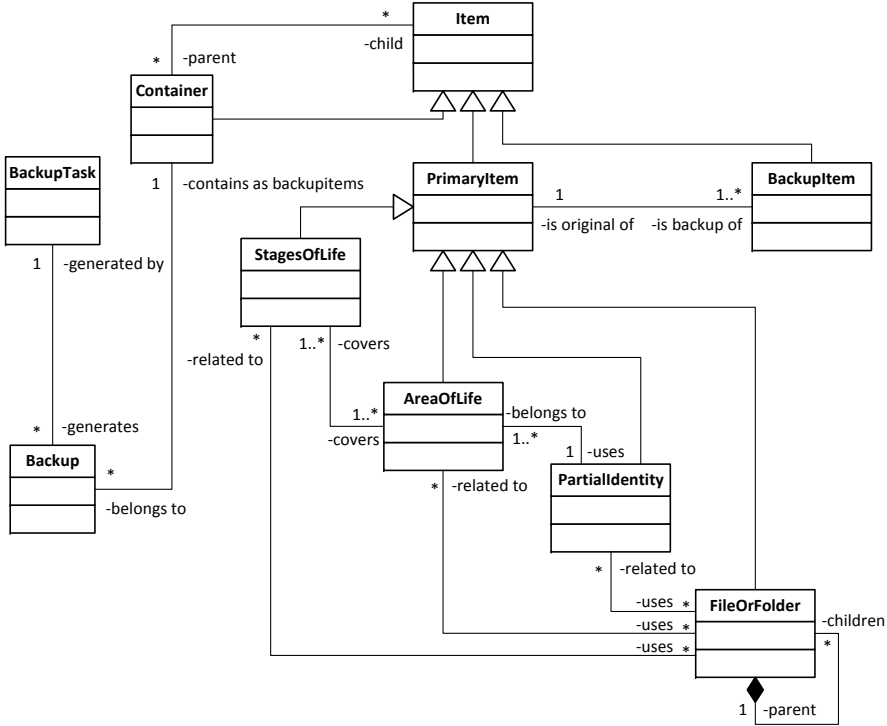


Fig. 4.3: Relations among of Areas of Life, partial identities, files etc.

### 4.2.1.3 Backup and Restore

“Backup” and “Restore” are the most prominent functionalities common to nearly any existing backup solution. Moreover, most backup tools operate on a *data* level, that is the user selects files, directories, partitions or whole disk drives. This kind of selection mode is supported in the PrimeLife backup demonstrator as well.

In addition to this common data level based selection mechanisms, the demonstrator offers an *identity* based selection mode. Remember that a basic concept of privacy-enhanced identity management is the separation into different partial identities, areas of life and stages of life. Thus, the user can select from each of these categories, e.g., specify which area(s) of life or partial identities he wants to backup.

Items of different types can be grouped together within a so-called container. A container can be understood as template of the primary items to be backed up. This would ease the related selection during the creation of a new backup task.

Areas of life, partial identities and files are related to each other (see [Figure 4.3](#)). An Area of Life typically covers multiple partial identities; likewise a partial identity is related to many files. Note that a file can be related to more than one partial

identity and area of life, e.g., a digital photo that shows persons from the “Family Area of Life” as well as the “Working Area of Life.”

In the field of privacy-enhanced identity management, one of the basic reasons for differentiating between different partial identities, areas of life etc. is to avoid unwanted linkability between them. Consequently, this unlinkability property has to be preserved for the backups as well. Thus, even if the user can group different partial identities or areas of life together for creating one logical backup task, the actual backup data need to be stored separately. Assume for instance, that the user selects files that belong to two different partial identities. In this case, two different backups will be created, ideally stored on two different online storage providers. Otherwise an attacker might learn that the two partial identities in question actually belong to one and the same user.

You might wonder how the backup demonstrator knows about the existing areas of life, partial identities and the relation among them and the files stored on the given machine. Conceptually, this information is provided as a core functionality of a privacy-enhanced identity management system, which in turn is out of scope of the backup demonstrator itself. Thus, for the backup demonstrator, we simply assume that such an IDM system is in place. However, it does not really exist today in practice. Therefore, we decided to create mockup data (areas of life, partial identities, files and folders) that are used for demonstrating the privacy-preserving backup aspects related to them. Nevertheless, the backup demonstrator allows for creating backups of real files and folders, but these items are not associated with any partial identity or area of life. Thus, from a privacy (linkability) point of view, they are treated as being “equal” and therefore are stored within a single backup.

#### 4.2.1.4 Delegation

Delegation is considered to be one of the most important aspects to be shown by the demonstrator. It cannot only be seen as an innovative feature in itself, which is not common in today’s backup tools, but it also has many implications with respect to the area of lifelong privacy, which in turn is the underlying motivation for the demonstrator.

From a functional point of view, delegation means that a primary user (the delegator) delegates access rights to a particular backup to some delegates (see [Figure 4.4](#)). These access rights are usually bound to a policy describing under which circumstances the access should be granted. A typical use case would be that a user delegates the access rights to the files related to his work partial identity/area of life to his colleagues under the policy that the access is only permitted if the user becomes ill. Moreover, the policy would express the obligation that the user needs to be informed if one of his colleagues really accesses the backup.

Delegation does not only deal with the privacy of the delegator but also with the privacy of the delegates. Therefore, a delegate will be asked if he is willing to accept the delegation request. Thereby in the spirit of informed consents he will be informed that each access to the backup would also be reported to the delegator.

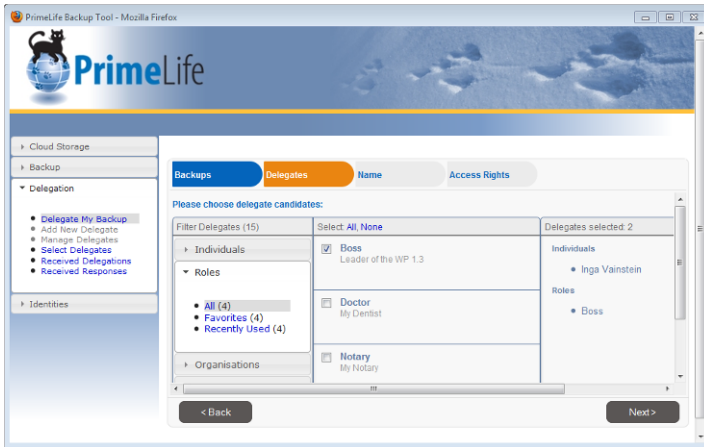


Fig. 4.4: Management console of the backup demonstrator showing the interface for selection of delegates.

It is currently an open question as to how much additional “meta-data” about a given backup will be communicated to a delegate. On the one hand, a delegate might want to know beforehand which files are included in a given backup, so that he can better decide if he really wants to accept the delegation or access the backup, respectively. On the other hand, this information might already have some negative impact on the privacy of the delegator. For enhancing the privacy of the delegator, it is desirable that a delegate only learns that information if he actually accesses a given backup. Thus, if the conditions defined in the access policy never become true, any unnecessary information flow will be avoided.

In the current version of the demonstrator, the list of possible delegates is predefined as mockup data. There are plans to integrate other sources of information for these address book-like data. Natural sources are social networks such as Facebook, Xing, LinkedIn etc. Moreover, the current demonstrator uses plain e-mail messages for transmitting delegations and the related acceptance responses. Future versions of the demonstrator might use the communication infrastructure offered by the mentioned social networks as well.

The delegation itself is an XML document describing the contents of the backup, the location of the backup and under which circumstances (policy) the backup can be accessed by the delegate. The delegation might also transfer some credentials (e.g., issued by the delegator) to the delegate, which the delegate will need in order to access the backup.

Conceptually, a lot of modern cryptographic mechanisms exist that could be used to construct a privacy-enhanced protocol for the purpose of delegation. Such a solution would require infrastructural support that is not in place today. Examples would be anonymous credentials issued by governmental or public institutions, public key infrastructures, attribute based access control mechanisms etc. Therefore we

decided to implement a much simpler scenario, which on the one hand is much closer to the current practice and on the other hand integrates components developed within the PrimeLife project and thus would not only demonstrate delegation itself, but also illustrate the interplay of the various components developed within the PrimeLife project.

Our delegation scenario comprises the following entities/components:

1. The eCV, a component which allows a user to store an electronic version of his CV. This component was developed within the PrimeLife project to demonstrate privacy aspects in service oriented architectures (see Section 21).
2. A trusted third party (TTP) utilising the PrimeLife policy engine (see Section 20).
3. A legacy online storage provider.
4. The delegator.
5. A delegate.

In this scenario, we demonstrate how a delegator can delegate the access to his backup to a delegate who can access the backup in case the delegator is ill. The delegation then would comprise the following steps:

1. The delegator stores the encrypted backup at the legacy online storage provider. The encryption is done using a symmetric cipher and the key  $k$ .
2. The delegator generates a random value  $k_1$  and calculates  $k_2$  such that  $k = k_1 \oplus k_2$ . Note that  $k_2 = k \oplus k_1$ , i.e.,  $k_2$  can be seen as a one time pad encryption of  $k$  using the key  $k_1$ .
3. The delegator stores  $k_1$  at the TTP. A PPL policy regulates the access to  $k_1$  saying that:
  - Only give access within a certain time frame and
  - to someone who can present a credential  $C_1$  and
  - a credential proving that the delegator is currently ill.
  - The obligation is to delete  $k_1$  after the time frame is over and
  - to inform the delegator if someone has accessed  $k_1$ .
4. The delegator sends  $k_2$ ,  $C_1$ ,  $C_2$  to the delegate (encrypted under the public key of the delegate). The delegator informs the delegate about the circumstances under which (illness of delegator and valid time frame) and how the delegate can access the backup.

Now let's assume that the delegator does in fact become ill. In this case, the delegator (or his doctor) sends a certification of illness to the eCV of the delegator. Moreover, the access policy to that certification says that someone who shows credential  $C_2$  is allowed to access the certificate of illness (in this case the delegator should be informed by the eCV).

In case the delegate wants to access the backup of the delegator and thus uses the rights delegated to him, the following steps happen:

1. The delegate downloads the encrypted backup. How the access control to the encrypted backup is done depends on the methods provided by the legacy online

- storage provider. Usually there exist some means of sharing the stored data with a defined set of users. But a broader discussion of this issue is out of scope here.
2. The delegate shows credential  $C_2$  to the eCV and requests the certificate of illness of the delegator. Note that the delegate has received  $C_2$  in step 4 during the delegation process.
  3. The delegate requests  $k_1$  from the TTP. He therefore shows credential  $C_1$  together with the certificate of illness of the delegator to the TTP. The delegate gets  $k_1$  and the TTP informs the delegator about that access to  $k_1$ .
  4. Now the delegate is able to calculate  $k = k_1 \oplus k_2$  and can decrypt the encrypted backup of the delegator.

Of course, the description above shows a very brief overview explaining the general ideas we have with respect to using existing components developed by the PrimeLife project for the backup demonstrator. Further research needs to be done to avoid/remove all the linkage that remains between the different steps (e.g., using different transaction pseudonyms for delegator and delegate etc.). Moreover, the information the various parties learn should be minimised (e.g., the TTP does not need to know that the “certificate of illness” is actually a certification of illness (e.g., we could use “meaningless” credentials here)). As a final remark, please note that all the parties mentioned above can be distributed (in the usual  $k$  out of  $n$  setting). To some extent, the involvement of the TTP already is a kind of distribution, because the eCV and the TTP can be seen as entities that store information accessible under a given access policy.

### 4.2.2 Deployment and Usage of the Demonstrator

Due to the platform independent design based on Java and web-technologies, the demonstrator can be installed and run on many modern operating systems including Linux, Mac OS X and Windows. Nevertheless, we decided to create a dedicated virtual machine based on VirtualBox<sup>15</sup> as virtual machine monitor and Ubuntu<sup>16</sup> Linux as guest operating system, which contains all the necessary components pre-installed. This makes the process of “playing” with the demonstrator much easier, especially if it comes to the more complex delegation scenarios. Besides the demonstrator itself, the virtual machine contains a local WebDAV online storage provider, two separate user accounts (one for the primary user/delegator and one for the delegate), a local e-mail infrastructure (SMTP and IMAP servers) etc. In order to make the installation of the virtual machine itself as easy as possible, a screencast explaining the necessary steps was created.

---

<sup>15</sup> <http://www.virtualbox.org/>

<sup>16</sup> <http://www.ubuntu.com/>



### 4.3 Concluding Remarks

In this chapter, the concept of the privacy-enhanced backup and synchronisation demonstrator was presented. It was shown that the objectives of lifelong privacy lead to practical results, which can be applied for solving real-life issues in enhanced ways. Our demonstrator reveals new problems that emerge as soon as lifelong aspects related to the data subject are taken into consideration. We presented a new approach, which can help the average citizen to protect himself against unwanted data loss respecting his different partial identities and areas of life. Our approach proceeds in such a way that it takes into account lifelong aspects of a human being and corresponding implications within the scope of the privacy.

Nevertheless, although for many of the envisaged problems that need to be solved with respect to lifelong aspects and here especially lifelong privacy solutions are known in principle. However, sometimes the concrete implementations that need to be realised in the demonstrator are currently still under development. This on the one hand explains why certain aspects and mechanisms were only described at a very high (abstract) level and defines on the other hand the research and development roadmap for the remaining five months planned for the finalisation of the demonstrator.

### 4.4 Acknowledgements

Many thanks are due to Hagen Wahrig and Katrin Borcea-Pfitzmann for their contribution to the design and implementation of the demonstrator and to Sebastian Clauß for his contribution to the requirements analysis. Our thanks also go to everyone in the PrimeLife project who took part in the scenario, requirements and demonstrator discussions, especially Rainer Böhme for his collection of demonstrator ideas from these discussions.

# References Part I

- [ACK<sup>+</sup>10] Claudio Agostino Ardagna, Jan Camenisch, Markulf Kohlweiss, Ronald Leenes, Gregory Neven, Bart Priem, Pierangela Samarati, Dieter Sommer, and Mario Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the PRIME Project. *Journal of Computer Security*, 18(1):123–160, 2010.
- [Ada99] Anne Adams. The implications of users’ privacy perception on communication and information privacy policies. In *In Proceedings of Telecommunications Policy Research Conference*, Washington, DC, 1999.
- [AG06] Allesandro Acquisti and Ralph Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *6th Workshop on Privacy Enhancing Technologies*, 2006.
- [BK09] Jo Bryce and Mathias Klang. Young people, disclosure of personal information and online privacy: Control, choice and consequences. *Inf. Secur. Tech. Rep.*, 14(3):160–166, 2009.
- [BM09] Suzy Bausch and Michelle McGiboney. News release: Social networks & blogs now 4th most popular online activity. [http://en-us.nielsen.com/content/dam/nielsen/en\\_us/documents/pdf/Press%20Releases/2009/March/Nielsen\\_Social\\_Networking\\_Final.pdf](http://en-us.nielsen.com/content/dam/nielsen/en_us/documents/pdf/Press%20Releases/2009/March/Nielsen_Social_Networking_Final.pdf), March 2009.
- [BMH05] Matthias Bauer, Martin Meints, and Marit Hansen. Structured Overview on Prototypes and Concepts of Identity Management Systems; FIDIS Del. 3.1. Available from [http://www.fidis.net/fileadmin/fidis/deliverables/fidis-wp3-del3.1.overview\\_on\\_IMS.final.pdf](http://www.fidis.net/fileadmin/fidis/deliverables/fidis-wp3-del3.1.overview_on_IMS.final.pdf) (letzter Abruf 09.02.2009), 2005.
- [Bri98] David Brin. *The Transparent Society: Will Technology Force Us to Choose Between Privacy and Freedom?* Perseus Publishing, 1998.
- [Bur09] Jörg Burger. Lügnerin! Betrügerin! <http://www.zeit.de/2009/53/Internetmobbing?page=all>, December 2009.
- [CDF<sup>+</sup>07] J. Callas, L. Donnerhackle, H. Finney, D. Shaw, and R. Thayer. RFC 4880 - OpenPGP Message Format. Technical report, Internet Engineering Task Force, November 2007.
- [Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
- [CHP<sup>+</sup>09] Sebastian Clauß, Marit Hansen, Andreas Pfitzmann, Maren Raguse, and Sandra Steinbrecher. Tackling the challenge of lifelong privacy. In *eChallenges*, October 2009.
- [CK01] Sebastian Clauß and Marit Köhntopp. Identity management and its support of multilateral security. *Computer Networks*, 37(2):205–219, October 2001.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.

- [Con09] G. Conti. Googling security; how much does google know about you? New York, Addison Wesley publishers, p. 91., 2009.
- [CvH02] Jan Camenisch and Els van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 21 – 30, 2002.
- [Dör08] Nicola Döring. Reduced social cues / cues filtered out. In N. C. Krämer, S. Schwan, D. Unz, and M. Suckfüll, editors, *Medienpsychologie. Schlüsselbegriffe und Konzepte*, pages 290–297, Stuttgart, 2008. Kohlhammer.
- [DPW09] Opinion 2/2009 on the protection of children’s personal data (general guidelines and the special case of schools). Art. 29 Data Protection Working Party, 2009. [http://ec.europa.eu/justice\\_home/fsj/privacy/docs/wpdocs/2009/wp160\\_en.pdf](http://ec.europa.eu/justice_home/fsj/privacy/docs/wpdocs/2009/wp160_en.pdf).
- [EB09] Lilian Edwards and Ian Brown. Data control and social networking: Irreconcilable ideas? In A. Matwyshyn, editor, *Harboring Data: Information security, law and the corporation*. Stanford University Press, 2009.
- [EGH08] Anja Ebersbach, Markus Glaser, and Richard Heigl. *Social Web*, volume 3065 of *UTB*. UVK, Konstanz, 2008.
- [EK02] Gunther Eysenbach and Christian Köhler. How do consumers search for and appraise health information on the world wide web? Qualitative study using focus groups, usability tests, and in-depth interviews, 2002.
- [ENI08] ENISA. Technology-induced Challenges in Privacy and Data Protection in Europe. A report by the ENISA Ad Hoc Working Group on Privacy and Technology, European Network and Information Security Agency (ENISA), Heraklion, Crete, Greece, October 2008.
- [Fac] Facebook statistics. <http://www.facebook.com/press/info.php?statistics>. last access 21 October 2010.
- [FC01] J. W. Fritch and R. L. Cromwell. Evaluating internet resources: identity, affiliation, and cognitive authority in a networked world. *Journal of the American Society for Information Science and Technology*, 52(6):499–507, 2001.
- [FM00] A. J. Flanagan and M. J. Metzger. Perceptions of internet information credibility. *Journalism & Mass Communication Quarterly*, 77(3):515–540, 2000.
- [FSD<sup>+</sup>03] B. J. Fogg, C. Soohoo, D. R. Danielson, L. Marable, J. Stanford, and E.R. Trauber. How do users evaluate the credibility of web sites? a study with over 2,500 participants. proceedings of dux2003, designing for user experiences conference. <http://www.consumerwebwatch.org/dynamic/web-credibility-reports-evaluate-abstract.cfm>, 2003.
- [GA05] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *WPES ’05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, New York, NY, USA, 2005. ACM.
- [Gof59] Erving Goffman. *The presentation of self in everyday life*. Doubleday, 1959.
- [Gri08] James Grimmelman. Facebook and the social dynamics of privacy [draft version]. [http://works.bepress.com/james\\_grimmelman/20/](http://works.bepress.com/james_grimmelman/20/), 2008.
- [HBPP05] Marit Hansen, Katrin Borcea-Pfitzmann, and Andreas Pfitzmann. PRIME – Ein europäisches Projekt für nutzerbestimmtes Identitätsmanagement. *it – Information Technology, Oldenbourg*, 6(47):352–359, 2005.
- [Hou09] Michelle G. Hough. Keeping it to ourselves: Technology, privacy, and the loss of reserve. *Technology in Society*, 31(4):406–413, 2009.
- [How08] Jeff Howe. *Crowdsourcing: Why the power of the crowd is driving the future of business*. Crown Business, 2008.
- [HPS08] Marit Hansen, Andreas Pfitzmann, and Sandra Steinbrecher. Identity management throughout one’s whole life. *Information Security Technical Report*, 13(2):83–94, 2008.
- [HRS<sup>+</sup>10] Marit Hansen, Maren Raguse, Katalin Storf, Harald Zwingelberg, Marit Hansen, Maren Raguse, Katalin Storf, and Harald Zwingelberg. Delegation for privacy management from womb to tomb – a european perspective. In Michele Bezzi, Penny Duquenoy, Simone Fischer-Hübner, Marit Hansen, and Ge Zhang, editors, *Privacy and Identity Management for Life*, volume 320 of *IFIP Advances in Information and Communication Technology*, pages 18–33. Springer Boston, 2010. 10.1007/978-3-642-14282-6\_2.

- [Ind08] Irish Independent. Taxman admits to Facebook 'trawl'. <http://www.independent.ie/national-news/taxman-admits-to-facebook-trawl-1297118.html>, February 2008.
- [Joi08] Adam N. Joinson. 'looking at', 'looking up' or 'keeping up' with people? motives and uses of facebook. In *CHI 2008*. ACM, 2008.
- [KPS11] Benjamin Kellermann, Stefanie Pötzsch, and Sandra Steinbrecher. Privacy-respecting reputation for wiki users. In *Proceedings of the 5th IFIP WG 11.11 International Conference on Trust Management (IFIPTM11)*, Copenhagen, Denmark, 2011.
- [LB08] Matthew M. Lucas and Nikita Borisov. Flybnight: Mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the Electronic Society (WPES)*. ACM, 2008.
- [Lea08] Charles Leadbeater. *We-think: Mass innovation, not mass production*. Profile, 2008.
- [Man10] Farhad Manjoo. Social networking your way to a new job. [http://www.nytimes.com/2010/08/26/education/26SOCIAL.html?pagewanted=1&\\_r=1](http://www.nytimes.com/2010/08/26/education/26SOCIAL.html?pagewanted=1&_r=1), August 2010.
- [Met07] M.J. Metzger. Making sense of credibility on the web: models for evaluating online information and recommendations for future research. *Journal of the American Society for Information Science and Technology*, 58(13):2078–2091, 2007.
- [MS09] Viktor Mayer-Schönberger. *Delete: The virtue of forgetting in the digital age*. Princeton University Press, 2009.
- [MSF09] Ines Mergel, Charlie Schweik, and Jane Fountain. The transformational effect of web 2.0 technologies on government. <http://ssrn.com/abstract=1412796>, 009.
- [New09] CBC News. Depressed woman loses benefits over Facebook photos. <http://www.cbc.ca/canada/montreal/story/2009/11/19/quebec-facebook-sick-leave-benefits.html>, November 2009.
- [O'R07] Tim O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. *Communications & Strategies*, 65(1):17–37, 2007.
- [PBP10] Stefanie Pötzsch and Katrin Borcea-Pfitzmann. Privacy-respecting access control in collaborative workspaces. In M. Bezzi et al., editor, *Privacy and Identity, IFIP AICT 320*, pages 102–111, Nice, France, 2010. Springer.
- [PD03] Leysia Palen and Paul Dourish. Unpacking 'privacy' for a networked world. In *Computer-Human Interaction (CHI) Conference 2003*, pages 129–137, 2003.
- [php] phpBB. Official website. <http://www.phpbb.com>.
- [Pöt09] Stefanie Pötzsch. Privacy awareness – a means to solve the privacy paradox? In *Proceedings of the IFIP/FIDIS Internet Security and Privacy Summer School*, 2009.
- [PRI] PRIME. Privacy and Identity Management for Europe. <https://www.PRIME-project.eu/>.
- [Pri10a] PrimeLife WP1.2. Privacy enabled communities. In Ronald Leenes Bibi van den Berg, editor, *PrimeLife Deliverable D1.2.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, April 2010.
- [Pri10b] PrimeLife WP1.2. Privacy-enabled communities demonstrator. In Stefanie Pötzsch, editor, *PrimeLife Deliverable D1.2.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, February 2010.
- [Pri10c] PrimeLife WP1.3. Towards a privacy-enhanced backup and synchronisation demonstrator respecting lifetime aspects. In Jaromír Dobiáš, Katrin Borcea-Pfitzmann, and Stefan Köpsel, editors, *PrimeLife Heartbeat H1.3.6*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, March 2010.
- [PWG10] Stefanie Pötzsch, Peter Wolkerstorfer, and Cornelia Graf. Privacy-awareness information for web forums: results from an empirical study. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 363–372, New York, NY, USA, 2010. ACM.
- [RG10] Kate Raynes-Goldie. Aliases, creeping, and wall cleaning: Understanding privacy in the age of facebook, 2010.
- [Rie02] S.Y. Rieh. Judgement of information quality and cognitive authority in the web. *Journal of the American Society for Information Science and Technology*, 53(2):145–161, 2002.

- [SGL06] Martin Szugat, Jan Erik Gewehr, and Cordula Lochmann. *Social Software schnell & kompakt*. entwickler.press, 2006.
- [Sol07] Daniel J. Solove. *The future of reputation: Gossip, rumor, and privacy on the Internet*. Yale University Press, 2007.
- [Tad10] Monika Taddicken. Measuring Online Privacy Concern and Protection in the (Social) Web: Development of the APCP and APCP-18 Scale. In *60th Annual ICA Conference (International Communication Association)*, Singapur., June 2010.
- [Tap09] Don Tapscott. *Grown up digital: How the Net generation is changing your world*. McGraw-Hill, 2009.
- [Tuf08] Zeynep Tufekci. Can you see me now? audience and disclosure regulation in online social network sites. *Bulletin of Science, Technology and Society*, 28(1):20–36, 2008.
- [Wik] List of social networking websites. [http://en.wikipedia.org/wiki/List\\_of\\_social\\_networking\\_websites](http://en.wikipedia.org/wiki/List_of_social_networking_websites).
- [Yok07] A.C. Yokum. I lost my job because of Facebook. [http://www.associatedcontent.com/article/353378/i\\_lost\\_my\\_job\\_because\\_of\\_facebook.html](http://www.associatedcontent.com/article/353378/i_lost_my_job_because_of_facebook.html), August 2007.
- [YQH09] Alyson L. Young and Anabel Quan-Haase. Information revelation and internet privacy concerns on social network sites: A case study of facebook. In *C&T '09*, pages 265–274. ACM, 2009.

## **Part II**

# **Mechanisms for Privacy**

## Introduction

Today's society places great demand on the dissemination and sharing of information. Such a great availability of data, together with the increase of the computational power available today, puts the privacy of individuals at great risk. The objective of the mechanisms activity is therefore to do novel research on the different open issues of the complex problem of guaranteeing privacy and trust in the electronic society. Chapter 5 focuses on privacy-enhancing cryptographic technologies that can be used in practice. The chapter presents anonymous credential schemas and their extensions along with cryptographic applications such as electronic voting and oblivious transfer with access control. Chapters 6 and 7 addresses mechanisms supporting the privacy of the users (transparency support tools, privacy measurement) and their electronic interactions. In particular, Chapter 6 illustrates a privacy-preserving secure log system as an example of a transparency supporting tool, and Chapter 7 focuses on trust and interoperable reputation systems. Chapter 8 investigates the problem of assessing the degree of protection offered by published data and of protecting privacy of large data collections that contain sensitive information about users. The chapter presents an information theoretic formulation of privacy risk measures and describes fragmentation-based techniques to protect sensitive data as well as sensitive associations. Chapter 9 addresses the problem of providing users with means to control access to their information when stored at external (possibly untrusted) parties, presenting new models and methods for the definition and enforcement of access control restrictions on user-generated data. The chapter illustrates a novel solution based on translating the access control policy regulating data into an equivalent encryption policy determining the keys with which data are encrypted for external storage. The solution is complemented by an approach based on two layers of encryption for delegating to the external server possible updates to the access control policy (without the need for the data owner to re-encrypt and re-upload resources).

## Chapter 5

# Cryptographic Mechanisms for Privacy

Jan Camenisch, Maria Dubovitskaya, Markulf Kohlweiss, Jorn Lapon, and Gregory Neven

**Abstract** With the increasing use of electronic media for our daily transactions, we widely distribute our personal information. Once released, controlling the dispersal of this information is virtually impossible. Privacy-enhancing technologies can help to minimise the amount of information that needs to be revealed in transactions, on the one hand, and to limit the dispersal, on the other hand. Unfortunately, these technologies are hardly used today. In this paper, we aim to foster the adoption of such technologies by providing a summary of what they can achieve. We hope that by this, policy makers, system architects, and security practitioners will be able to employ privacy-enhancing technologies.

### 5.1 Introduction

The number of professional and personal interactions we are conducting by electronic means is increasing daily. These on-line transactions range from reading articles, searching for information, buying music, and booking trips, to peer-to-peer interactions on social networks. Thereby, we reveal a plethora of personal information not only to our direct communication partners but also to many other parties of which we are often not even aware. At the same time, electronic identification and authentication devices are becoming more and more widespread. They range from electronic tickets and toll systems, to eID cards and often get used across different applications.

It has become virtually impossible to control where data about us are stored and how they are used. This is aggravated as storage becomes ever cheaper and the fact that the increasingly sophisticated data mining technologies allow for all of these data to be used in many ways that we can not even imagine today.

It is thus of paramount importance to enable individuals to protect their electronic privacy. Luckily, there exists a wide range of privacy enhancing technologies available that can be used to this end. These range from privacy-aware access control



and policy languages to anonymous communication protocols and anonymous credential systems. The PRIME (Privacy-Enhancing Identity Management for Europe) project [PRib] has shown that these technologies can indeed be used together to build trust and identity management systems that allows for protecting one's on-line privacy and that they are ready to be applied in practice. The PrimeLife project [pria] has taken up these research results and is concerned with bridging the gap from research to practice.

Let us, however, note that while technology can help, users also need to learn about the perils of our digital world and how to guard their privacy. Of course, ICT systems must, to this end, provide sufficient information to the users about what is happening with their data.

It seems that making use of privacy-enhancing technologies is harder than for other security technologies. One reason for this might be that the properties that they achieve are often counter-intuitive, in particular in cases of cryptographic building blocks. In an attempt to foster the adoption of privacy-enhancing technologies (PETs), we overview in this paper the most important cryptographic PETs and summarise what they achieve. We also give references for their technical details. Finally, we explain how these technologies can be embedded into larger systems.

## 5.2 Cryptography to the Aid

There is a large body of research on specific cryptographic mechanisms that can be used to protect one's privacy. Some of them are theoretical constructs, but many are actually fully practical and can be readily applied in practice. We here concentrate on the latter ones.

The oldest types of privacy-protecting cryptography are of course encryption schemes by themselves: they allow one to protect information from access by third parties when data is stored or sent to a communication partner. There are, however, a number of variants or extensions of such basic encryption that have surprising properties that can offer better protection in many use cases as we shall see. Apart from encrypting, one often needs to authenticate information. Typically, this is done by using a cryptographic signature scheme. The traditional signature schemes typically provide too much authentication in the sense that they are used in a ways that reveals a lot of unnecessary contextual information. The cure here is offered by so-called anonymous credential schemes and their extensions which we will present. Finally, we briefly discuss a number of cryptographic applications such as electronic voting schemes and privacy-enhanced access control schemes.

### 5.3 Private Credentials, Their Extensions, and Applications

Certified credentials form the cornerstones of trust in our modern society. Citizens identify themselves at the voting booth with national identity cards, motorists demonstrate their right to drive cars with driver licenses, customers pay for their groceries with credit cards, airline passengers board planes with their passports and boarding passes, and sport enthusiasts make their way into the gym using their membership cards. Often such credentials are used in contexts beyond what was originally intended: for example, identity cards are also used to prove eligibility for certain social benefits, or to demonstrate being of legal age when entering a bar.

Each of these credentials contains attributes that describe the owner of the credential (e.g., name and date of birth), the rights granted to the owner (e.g., vehicle class, flight and seat number), or the credential itself (e.g., expiration date). The information in the credentials is trusted because it is certified by an issuer (e.g., the government, a bank) who in its turn is trusted.

There are a number of different ways how such credentials can be technically realised. Depending on their realisation, they offer more or less protection of the user's privacy. For instance, they are often realised by an LDAP directory maintained by the issuer. That means that a user who wants to use a credential with a particular party (the verifier), will have to authenticate, typically with a username and password, towards the verifier who will then look up the user's credentials in the LDAP directory. While this realisation might satisfy the security requirement of the verifier and the issuer, it offers virtually no protection to the users. Apart from username/password being a rather insecure authentication mechanism, the user has 1) no control over which information the verifier requests from the issuer and 2) the issuer learns with which verifier the user is communicating.

A better realisation of credentials is with certificates with so-called attribute extensions [CSF<sup>+</sup>08]. Here, the user chooses a public/secret key pair and then obtains a certificate from the issuer on her public key. The certificate includes all statements that the issuer vouches for about the user. The user can then send this certificate to the verifier together with a cryptographic proof of ownership of the secret key. The user knows which data is revealed to the verifier by the certificate, but has to reveal all of the included attributes so that the verifier can check the issuer's signature. Moreover, if the verifier and the issuer compare their records, they can link the user's visit to the issuing of the credential by simply comparing the issuer's signature.

Anonymous credentials [Cha81, Bra99, CL01] (often also called private credentials or minimal disclosure tokens) solve all these problems and indeed offer the best privacy protection possible while offering the same cryptographic security. They work quite similarly to attribute certificates, the difference being that they allow the user to "transform" the certificate into a new one containing only a subset of the attributes of the original certificate. This feature is often called *selective disclosure*. The issuer's signature is also transformed in such a way that the signature in the new certificate cannot be linked to the original signature; this is usually called *unlinkability* in the literature.

### 5.3.1 *Extended Functionalities*

Apart from the basic features of selective disclosure and unlinkability sketched above, many anonymous credential systems offer additional features that can be very useful in practical use cases. In the following, we discuss the most important of these features.

#### Attribute Properties

Rather than revealing the complete value of an attribute, some credential systems allow the user in the transformation to apply any (mathematical) function to the original attribute value. For instance, if the original certificate contains a birthdate, the transformed attribute could contain only the decade in which the user was born. As a special case, the function could be boolean (meaning, having as output “true” or “false”), so that only the truth of a statement about the attribute is revealed. For instance, based on the birthdate in a certificate, the user could prove that she is between 12 and 14 years old. The schemes also allow for logical AND and OR combinations of such boolean expressions [CDS94].

#### Verifiable Encryption

This feature allows one to prove that a ciphertext encrypts a value that is contained in a credential. For instance, a service provider could offer its service to anonymous users provided that they encrypt their name and address, as contained in their identity card, under the public key of a trusted third party, such as a judge. The cryptography ensures that the service provider himself cannot decrypt the name and address, but can rest assured that the ciphertext contains the correct value. In case of misuse of the service, the service provider or a law enforcement agency can then request the third party to decrypt the user’s name and address from the ciphertext, i.e., to revoke the anonymity of the user. Note that it can be decided at the time of showing the credential, whether or not any information in the credential should be verifiably encrypted, i.e., this need not be fixed at the time the credential is issued and can be different each time a credential is shown.

An essential feature that we require in this setting from an encryption scheme is that of a label [CS03]. A label is a public string that one can attach to a ciphertext such that without the correct label, the ciphertext cannot be decrypted. The most common usage for the label in our setting is to bind the conditions and context under which the trusted third party is supposed to decrypt (or not decrypt) a given ciphertext.

In principle, one can use any public encryption scheme for verifiable encryption [CD00]. The most efficient way to do so, however, is probably using the Paillier encryption scheme [Pai99] for which efficient proof protocols exist with a variant, secure against chosen-ciphertext attacks [CS03]. Security against chosen-ciphertext

attacks is actually crucial in this setting: the trusted third party's job is essentially a decryption oracle and hence semantic security is not sufficient.

### Revocation of Credentials

There can be many reasons to revoke a credential. For example, the credential and the related secret keys may have been compromised, or the user may have lost her right to carry the credential. Also, sometimes a credential might only need to be partially revoked. For instance, an expired European passport can still be used to travel within Europe, or a driver's license revoked because of speeding could still be valid to prove the user's age or address.

Possible solutions for revocation in the case of non-anonymous credentials is to "blacklist" all serial numbers of revoked credentials in a so-called *certificate revocation list* [CSF<sup>+</sup>08] that can be queried on- or off-line. Another option is to limit the lifetime of issued credentials by means of an expiration date and periodically re-issue non-revoked credentials. The latter solution works for anonymous credential as well, although re-issuing may be more expensive than for ordinary credentials. The former solution as such does not work, as revealing a unique serial number of a credential would destroy the unlinkability property. However, the general principle of publishing a list of all valid (or invalid) serial numbers can still work if, rather than revealing their serial number, users leverage the attribute property feature to prove that it is among the list of valid serial numbers, or that it is not among the invalid ones. A number of protocols that work along these lines have been proposed [BS04, BDD07, NFHF09] where the solution by Nakansihi et al. [NFHF09] seems to be the most elegant one.

Another solution inspired by revocation lists is the use of so-called dynamic accumulators [CL02, CKS09]. Here, all valid serial numbers are accumulated (i.e., compressed) into a single value that is then published. In addition, dynamic accumulators provide a mechanism that allows the user to prove that the serial number of her credential is contained in the accumulated value. Whenever a credential is revoked, a new accumulator value is published that no longer contains the revoked serial number. The schemes, however, require that users keep track of the changes to the accumulator to be able to perform their validity proofs.

We observe that enabling revocation brings along the risk that the authority in control of the revocation list (or accumulator value) modifies the list to trace transactions of honest users. For instance, the authority could fraudulently include the serial number of an honest user in the revocation list and then check whether the suspected user succeeds in proving that her credential is not on the list. Such behaviour could of course be noted by, e.g., a consumer organisation monitoring changes to the public revocation values.

One idea to lessen the trust that one has to put into such a third party is by using threshold cryptography, i.e., by distributing the power to update the revocation list over multiple entities such that a majority of them is needed to perform an update.

### Limited-use credentials

Some credentials, such as entrance tickets, coupons, or cash money, can only be used a limited number of times. A very basic example of such credentials in the digital world is anonymous e-cash, but there are many other scenarios. For instance, in an anonymous opinion poll one might have to (anonymously) prove ownership of an identity credential, but each credential can only be used once for each poll. Another example might be an anonymous subscription for an on-line game, where one might want to prevent that the subscription credential is used more than once simultaneously, so that if you want to play the game with your friends, each friend has to get their own subscription [CHK<sup>+</sup>06].

When implementing a mechanism to control the number of times that the same credential can be used, it is important that one can define the scope of the usage restriction. For instance, in the opinion poll example, the scope is the specific poll that the user is participating in, so that participating in one poll does not affect his ability to participate in another one. For electronic cash, on the other hand, the scope is global, so that the user cannot spend the same electronic coin at two different merchants. Overspending occurs when the same credential is used more than specified by the usage limit within the same scope. Possible sanctions on overspending could be that the user is simply denied access to the service, or that some further attributes from the user's credential are revealed [CHL06, CHK<sup>+</sup>06].

With limited-use credentials, one can prevent users from sharing and redistributing their credentials to a large extent. Another means of sharing prevention is the so-called all-or-nothing sharing mechanism [CL01]. This mechanism ensures that if a user shares one credential with another user (which requires revealing to the other user the secret key material of that credential) then the other user can also use all the other credentials (because they are based on the same secret key material). In this case sharing a single credential would mean sharing one's entire digital identity, e.g., including access to one's bank account, which people probably are not prepared to do. If, however, one wishes to make sharing of credentials infeasible, then they need to be protected by tamper-resistant hardware, which we discuss next.

### Hardware Protection

Being digital, anonymous credentials are easily copied and distributed. On the one hand, this is a threat to verifiers as they cannot be sure whether the person presenting a credential is the one to whom it was issued. On the other hand, this is also a threat to users as it makes their credentials vulnerable to theft, e.g., by malware.

One means to counter these threats is to protect a credential by a tamper-resistant hardware device such as a smart card, i.e., to perform all operations with the credential on the device itself. A straightforward way of doing so in a privacy-friendly way would be to embed the same signing key in all issued smart cards. The disadvantage of this approach is that if the key of one card is compromised, all smart cards have to be revoked.

A more realistic approach is to implement the Camenisch-Lysyanskaya credential system on a standard Java card [BCGS09]. However, depending on the type of smart card, it might only be possible to process a single credential on the device. In this case, one could still bind other credentials to the device by including in each credential an identifier as an attribute that is unique to the user [Cam06]. All of a user's credentials should include the same identifier. (The issuing of these credentials can even be done without having to reveal this identifier.) When an external credential (i.e., a credential that is not embedded in the smart card) is shown, the verifier requires the user to not only show the external credential but also the credential on the smart card, together with a proof that both credentials contain the same identifier. Using the attribute properties feature, users can prove that both credentials contain the same identifier without revealing the identifier.

### ***5.3.2 Direct Anonymous Attestation***

How can a verifier check that a remote user is indeed using a trusted hardware module, without infringing on the privacy of the user, and without having to embed the same secret key in each module? This question arose in the context of the Trusted Computing Group (TCG). In particular, the Trusted Platform Module (TPM) monitors the operating system and can then attest to a verifier that it is pristine, e.g., free of viruses and thus safe for running an application such as e-banking. To protect privacy, the TCG has specified a scheme for this attestation that can essentially be seen as a group signature scheme without the opening functionality, so that anonymity cannot be revoked [BCC04] but with a revocation feature such that stolen keys can nevertheless be identified and rejected.

## **5.4 Other Privacy-Enhancing Authentication Mechanisms**

There are a number of primitives that are related to anonymous credentials. Some of them are special cases of anonymous credentials, while others can be seen as building blocks or share the same cryptographic techniques to achieve anonymity.

### **Blind Signatures**

A blind signature scheme [Cha83] allows a user to get a signature from the signer without the signer being aware of the message nor the resulting signatures. Thus, when the signer at some later point is presented with a valid signature on a message, he is not able to link it back to the signing session that produced the signature. Blind signature schemes are a widely used building block for schemes to achieve anonymity. Examples include anonymous electronic voting [Cha83, FOO91] and

electronic cash [Cha83], which we discuss below. A large number of different blind signature schemes have been proposed in the literature based on various cryptographic assumptions; there are too many to be listed here.

The main feature of blind signatures is that the signer has no control whatsoever on the message being signed. This feature can at the same time be a drawback. Typically, the signer wants to impose certain restrictions on the message that he's signing, such as the expiration date of a credential, or the denomination of a digital coin. When used in protocols, blind signatures therefore often have to be combined with inefficient "cut-and-choose" techniques, where the user prepares many blinded versions of the message to be signed, all but one of which are to be opened again, and the remaining one is used to produce the signature. A more efficient approach is to use *partially* blind signatures [AF96], where the signer determines part of the signed message himself, allowing him to include any type of information, such as the issuance or expiration date of the signature.

## Electronic cash

The goal of (anonymous) electronic cash [Cha83] is to prevent fraud while achieving the same privacy guarantees as offered by cash money in the real world. In particular, when a user withdraws an electronic coin from the bank, spends it at a merchant, and the merchant deposits the electronic coin at the bank, the bank cannot link the coin back to the user. However, if either the user or the merchant try to cheat by spending or depositing the same coin twice, the identity of the fraudster is immediately revealed.

Online electronic cash, i.e., where the bank is online at the moment a coin is spent, can be built using blind signatures by having the bank blindly sign random serial numbers. After having issued the blind signature to a user, the bank charges the user's account. The user can spend the money with a merchant by giving away the random serial number and the signature. To deposit the coin, the merchant forwards the serial number and signature to the bank, who verifies the signature and checks whether the serial number has been deposited before. If not, the bank credits the merchant's account; if so, the bank instructs the merchant to decline the transaction.

In off-line electronic cash [CFN88], the bank is not involved when the coin is spent, only when it is withdrawn or deposited. The techniques described above are therefore enhanced to, at the time of deposit, distinguish between a cheating user and a cheating merchant, and in the former case, to reveal the identity of the cheating user. Both online and off-line electronic anonymous cash can be seen as special cases of limited-use anonymous credentials as described above, where a single scope is used for all payments. To obtain off-line electronic cash, the user is required to provide a verifiable encryption of her identity, which is only decrypted in case of fraud.

## Group Signatures

A group signature scheme [CvH91] allows group members to sign messages in a revocably anonymous way, meaning that any verifier can tell that the message was signed by a group member, but not by which group member, while a dedicated opening manager can lift the anonymity of a signature and reveal the identity of the signer who created it. Group membership is controlled by a central group manager, who generates the group's public key and provides the individual members with their secret signing keys. Some schemes combine the roles of group manager and opening manager in a single entity.

Group signatures satisfy a whole range of security properties, including unforgeability (i.e., no outsider can create valid signatures in name of the group), unlinkability (i.e., signatures by the same signer cannot be linked), anonymity (i.e., nobody except the opening manager can tell which signer created a signature), traceability (i.e., any valid signature can be traced back to a signer), exculpability (i.e., no collusion of cheating signers can create a signature that opens to an honest signer), and non-frameability (i.e., not even a cheating group manager can create a signature that opens to an honest signer). Many of these properties are in fact related [BMW03, BSZ05].

The showing protocol of many anonymous credential systems follows a typical three-move structure that allows them to be easily converted into a signature scheme by means of a hash function [FS87]. The resulting signature scheme inherits all the anonymity features of the credential system. A group signature scheme can then be obtained by combining it with verifiable encryption: the issuer plays the role of group manager and issues to each group member a credential with a single attribute containing his identity. Group members do not reveal their identity attribute when signing a message, but verifiably encrypt it under the public key of the opening manager. One can take this approach even further by including more attributes and using the attribute properties feature. For example, one could create a signature that reveals that one authorised group member between 18 and 25 years old signed the message, but only the opening manager can tell who exactly did.

## Ring Signatures

One possible disadvantage of group signatures is that the group manager decides on the composition of the group, and that members can only sign in the name of that group. Ring signatures [RST01] are a more flexible variant of group signatures that have no group manager or opening manager. Rather, users can determine the group of “co-signers” at the time a signature is created. The co-signers’ collaboration is not needed in the signing process, so in fact, they need not even be aware that they are involved in a ring signature. There is no authority to reveal the identity of the signer behind a ring signature, but some schemes allow the signer to voluntarily prove that they created a signature.



## Redactable and Sanitisable Signatures

In some applications, it may be necessary to hide words, sentences, or entire paragraphs of a signed document without invalidating the original signature. This is exactly what redactable [JMSW02] and sanitisable [ACdMT05] signatures allow one to do, the difference being that in the former anyone can censor a document, while in the latter only a censoring authority designated by the original signer can do so. Both primitives satisfy a privacy property implying that it is impossible to link back a censored signature to the original signature that was used to create it.

### *5.4.1 Privacy-Enhancing Encryption*

While the main focus of this work is on privacy-enhancing authentication, a complete privacy-friendly infrastructure also involves special encryption mechanisms. We already touched upon verifiable encryption in relation to anonymous credentials. We discuss a selection of other privacy-relevant encryption primitives here.

#### Anonymous Communication

Most of the anonymous authentication mechanisms described above rely on an anonymous underlying communication network: cryptographic unlinkability of signatures clearly does not help if the users are identifiable by their IP address. Mix networks [Cha81] can be used to obfuscate which user communicates with which servers by routing the traffic through an encrypted network of mix nodes. The exact route that a packet follows can either be decided by the mix node or by the sender of the packet. In the latter case, the message is wrapped in several layers of encryption, one layer of which is peeled off at each node; this process is often referred to as onion routing [Cha81, GRS99, CL05]. So-called dining cryptographer networks or DC-nets [Cha88] even hide the fact whether entities are communicating at all, but they of course incur a constant stream of dummy traffic between all participants in doing so.

#### Homomorphic and Searchable Encryption

With current technology trends such as software as a service and cloud computing, more of our information is stored by external services. Storing the information in encrypted form is often not an option, as it ruins either the service's functionality or its business model. As the main goal of encryption is to hide the plaintext, it usually destroys any structure present in the plaintext; tampering with a ciphertext either renders it invalid, or turns the plaintext into unpredictable random garbage. Some encryption algorithms however are homomorphic, in the sense that applying

certain operations on ciphertexts has the effect of applying other operations on the plaintexts. One can thereby process encrypted data without decrypting it, so that for example a server can apply data mining mechanisms directly on encrypted information [OS07]. There exist homomorphic encryption schemes that support multiplication [ElG85] and addition [Pai99] of plaintexts, and since recently, also schemes that support both at the same time [Gen09].

In similar scenarios it can be useful if a server can search through encrypted information without having to decrypt it. For example, this would enable an encrypted email hosting server to perform efficient searches on your email and transmit only the matching (encrypted) emails. Special-purpose schemes have been developed for this purpose as well, both in the symmetric [SWP00] and the asymmetric [BCOP04] setting.

### Oblivious Transfer

Imagine a database containing valuable information that is not sold as a whole, but that rather charges customers per accessed record. At the same time, the list of queried records reveals sensitive information about the customers' intentions. For example, a company's search queries to a patent database or to a DNA genome database may reveal its research strategy or future product plans.

An oblivious transfer protocol [Rab81] solves this apparently deadlocked situation by letting a client and server interact in such a way that the server does not learn anything about which record the client obtained, while the client can only learn the content of a single record. The adaptive variant [NP99] of the primitive can amortise communication and computation costs over multiple queries on the same database.

## 5.5 Electronic Voting, Polling, and Petitions

Voting privacy is more than just a desirable feature, it is a fundamental principle for a democratic election. Electronic voting schemes have been proposed based on mix networks [Cha81], based on homomorphic encryption [CF85], and based on blind signatures [FOO92]. Electronic voting schemes form the backbone of e-democracy and should be properly designed and verified to guarantee a variety of security properties, such as end-to-end verifiability, voter anonymity, as well as coercion and receipt freeness.

Other mechanisms such as electronic petitions and verifiable electronic opinion polls aim at strengthening participatory democracy. The limited-use restrictions of anonymous credentials makes them applicable to such scenarios. As discussed in Section 5.3.1, the scope of an anonymous credential with limited show in an opinion poll or e-petition system is the identifier of the poll/petition that the user wants to participate in. In this way, anonymous credentials with limited show function-

ality restrict a user to signing only once for a specific poll, without affecting her possibility to participate in other polls/petitions.

Several demonstrators based on this basic idea have been built to show the feasibility of this approach [DKD<sup>+</sup>09, BP10, VLV<sup>+</sup>08, TBD<sup>+</sup>10]. The latter also implemented parts of the anonymous credential protocol in a chip card using software technology and hardware[jav10] similar to the one used in European identity cards. The deployment of such a system would bind an electronic petition signing directly to a European citizen. The properties of the anonymous credential system would allow for further restrictions to the scope of the partition. For instance, for local issues it would be required to be a resident of a particular district in order to be able to participate in the petition. Moreover, with this technology, it is simple to extend the application such that a poll may include restrictions on who can participate (e.g., only persons older than 18). Optionally, the user may selectively disclose attributes or properties of those attributes (e.g., an age interval) that may be used for statistics.

## 5.6 Oblivious Transfer with Access Control and Prices

The techniques described above can be combined in various way to address interesting business needs. For example, imagine that each record in a patent or DNA database as described above is protected by a different access control policy, describing the roles or attributes that a user needs to have in order to obtain it. By combining anonymous credentials with adaptive oblivious transfer protocols, one can construct solutions where the user can obtain the records she is entitled to, without revealing the applicable access control policy to the database, or which roles she has [CDN09]. By another combination of such techniques, the database can attach different prices for each record, and let users only download as many records as their prepaid balance allows, all while remaining completely anonymous [CDN10].

### Oblivious Transfer with Access Control

Consider the case of access to a database where the different records in the database have different access control conditions. These conditions could be certain attributes, roles, or rights that a user needs to have to access the records. The assigning of attributes to users is done by a separate entity called the issuer, external to the database. To provide the maximal amount of privacy, a protocol is required such that:

- Only users satisfying the access conditions for a record can access that record;
- The service (database) provider does not learn which record a user accesses;
- The service (database) provider shall not learn which attributes, roles, etc. a user has when she accesses a record, i.e., access shall be completely anonymous, nor shall it learn which attributes the user was required to have to access the record.

To fulfill all the above requirements, we construct an Oblivious Transfer with Access Control (AC-OT) protocol [CDN09], which is based on the oblivious transfer protocol by Camenisch et al. [CNS07] and works as follows. Each record in the database has an access control list (ACL). The ACL is a set of categories. We note that the name “category” is inspired by the different data categories that a user is allowed to access. However, the category could just as well encode the right, role, or attribute that a user needs to have in order to access a record.

The database server first encrypts each record with a unique key and publishes these encryptions. The encryption key is derived from the index of the record, the ACL of the record, and a secret of the database server. Although the secret of the database is the same for all record keys, it is not possible to derive the encryption key for one record from that of another record. Thus, to decrypt a record the user needs to retrieve the corresponding key from the server.

To be able to do this, the user has to obtain the necessary credentials from the issuer. Each anonymous credentials [Cha85, LRSW99, CL01], issued to a user, certifies a *category* of records the user is allowed to access. Recall that anonymous credentials allow the user to later prove that she possesses a credential without revealing any other information whatsoever. Also, anonymous credential systems provide different revocation mechanisms. Note that if a record has several categories attached to it, then the user must have a credential for *all* of these categories, basically implementing an AND condition. If one would want to specify an OR condition, one could duplicate the record in the database with a second set of categories.

To obviously access a record for which the user has the necessary credentials, she engages in a transfer protocol with the database and while retrieving a key, gives a zero-knowledge proof of knowledge that she possess credentials on all the categories that are encoded into the key that she wants to retrieve. If she succeeds then she can decrypt that record, otherwise, she cannot. The database learns nothing about the index of the record that is being accessed, nor about the categories associated to the record.

### Priced Oblivious Transfer with Rechargeable Wallets

Now consider a database where each record may have a different price, for example, DNA or patent database, as described above. In this setting, it is necessary to prevent the database from gathering information about a customer’s shopping behaviour, while still allowing it to correctly charge customers for the purchased items.

To solve this problem we propose the first truly anonymous priced oblivious transfer protocol (POT) [CDN10], where customers load money into their pre-paid accounts, and can then start downloading records so that:

- The database does not learn which record is being purchased, nor the price of the record that is being purchased;
- The customer can only obtain a single record per purchase, and cannot spend more than his account balance;
- The database does not learn the customer’s remaining balance; and

- The database does not learn any information about who purchases a record.

We note that previous POT protocols ([AIR01, Tob02, RKP09]) do not provide full anonymity (the last requirement) for the customers: the database can link transactions of the same customer. Furthermore, they also lack a recharge functionality: once a customer's balance does not contain enough credit to buy a record, but is still positive, the customer cannot use up the balance, but will have to open a new account for further purchases. Even if the protocol can be extended so that the customer can reveal and reclaim any remaining credit, he will leak information about his purchases by doing so. In our protocol, customers can recharge their balances anonymously at any time.

In addition, we provide an enhanced protocol where records are transferred using an optimistic fair exchange protocol [ASW97, ASW00], thereby preventing a cheating database from decreasing a customer's wallet without sending the desired record.

Here, in Priced Oblivious Transfer with Rechargeable Wallets protocol, as with the AC-OT protocol, the database provider encrypts and publishes the entire encrypted database. Each record is encrypted with a unique key that is derived from its index and its price.

To be able to access records, a customer first contacts the provider to create a new, empty wallet. Customers can load more money into their wallet at any time, using an anonymous e-cash scheme, for example.

When a customer wants to purchase a record with index  $i$  and price  $p_i$  from the database, the provider and the customer essentially run a two-party protocol, at the end of which the customer will have obtained the decryption key for the record  $i$  as well as an updated wallet with a balance of  $p_i$  units less. This is done in such a way that the provider does not learn anything about  $i$  or  $p_i$ . More precisely, we model wallets as one-time-use anonymous credentials with the balance of the wallet being encoded as an attribute. When the customer buys a record (or recharges her wallet), she basically uses the credential and gets in exchange a new credential with the updated balance as an attribute, without the provider learning anything about the wallet's balance. The properties of one-time-use credentials ensure that a customer cannot buy records worth more than what she has (pre-)paid to the provider.

To sum up, we construct protocols that allow users to obtain records from the database they are entitled to (by the access control rules and/or by purchasing a record), and at the same time provide full anonymity for the users and prevent the database from gathering any statistics about their transactions.

## 5.7 Oblivious Trusted Third Parties

Anonymous/private credentials allow for implementing electronic transactions that are unlinkable and for selectively disclosing the minimal amount of information about the user. At the same time, these transactions have to be accountable. When using anonymous credentials, transactions are automatically accountable in the

sense that the verifier is ensured that what is being proven during the credential show, is indeed vouched for by the issuer. However, many real-life applications have to consider exceptional cases in which additional information is required in case of a malicious transaction.

When the conditions for detecting such abuse can be expressed mathematically and can be detected inside of the electronic system, one can often mitigate such malicious transactions cryptographically. Examples for such transactions are offline double spending and money laundering resistant e-cash systems as well as the e-petition system sketched above.

In other situations, e.g., when a suspect might have used an anonymous credential for physical access control to a crime scene, the evidence that additional information is allowed to be recovered, e.g, the identity of all users that accessed the premise during a certain time period, lies outside of the system. The most simple solution is to reveal a verifiable encryption of this information during the show of the credential.

In particular, a user  $U$  would encrypt her true identity with the public key of the anonymity revocation authority  $RA$ , a form of trusted third party (TTP) and provides this encrypted data to a service provider  $SP$ . She then convinces  $SP$  in a zero-knowledge proof of knowledge that this encrypted data contains her valid user identity that can be opened by the authority if it decides that the opening request is legitimate.

This solution, however, raises several concerns.

1. It involves a fully trusted party, the revocation authority, that is able to link all transactions with no graceful degradation of privacy and security, should the revocation authority become compromised.
2. Additionally, the solution does not provide the best achievable accountability properties, as especially powerful users could bribe or threaten the  $RA$  such that it would refuse to open particular ciphertexts.
3. Honest service providers find the traditional system encumbering because of the need to involve such highly trusted authorities for even minor dispute cases. For example, to bring a case to law enforcement in the real world is likely to have a non-trivial cost, both in the time required, and in support from legal council.

There are two avenues that can be followed to reduce the trust into a trusted third party (TTP) like the revocation authority. One is to distribute the TTP such that it does not run on a single machine but on multiple machines. Each machine is owned by an organisation that is unlikely to collaborate against the user with the other organisations (e.g., a privacy office, the interior ministry, and the justice ministry). The cryptographic protocol that replaces the TTP guarantees that as long as one of these multiple machines is uncompromised and operates correctly, the other machines cannot infringe the user's privacy.

## Oblivious Anonymity Revocation

The other approach that we describe here is to design the protocol in such a way that the TTP is as oblivious as possible to the task it performs, e.g., it does not know which user's identity it helps to reveal: in our implementation the identity of the user would be protected by two layers of encryption. The revocation authority can only remove the outer layer of encryption. The second layer is removed by the service provider himself once he receives the partial decryption from the revocation authority.

This Oblivious Trusted Third Parties (OTTP) mechanism helps to achieve some amount of graceful degradation. Even if the revocation authority is compromised, it cannot learn any useful information. Here we assume that there is no collaboration between the service provider and the revocation authority.

Another aspect in which the revocation authority can be made oblivious is in terms of the information it receives from the service provider. We want to make sure that the original ciphertexts are labeled with the revocation condition but are otherwise only known to the service provider, i.e., they look random to all possible collusions between users and the revocation authority. This guarantees that powerful users with special interests have no way of influencing the revocation authority to selectively open only some of the opening requests.

In contrast to the fully trusted third party as discussed above, this scheme alleviates the trust assumptions on the TTP, and provides both stronger privacy and stronger accountability. The OTTP revocation authority is a weaker TTP, whose only trust requirement is to revoke the anonymity of users only in those situations in which the revocation condition indeed holds. To achieve this, the scheme restricts the revocation authority to only process blinded information, unknown to users, and to output blinded information that can only be decrypted by the service provider.

As a result, RA cannot block requests of SP selectively and cannot collude against any specific user, nor can it link the transactions of users in the system. Furthermore, a compromised authority remains restricted in the information it could possibly gather, i.e., it can only gather information if the service provider of a particular transaction consents to remove the remaining blinding.

Essentially, oblivious anonymity revocation resolves most of our concerns stated above. Nevertheless, in many scenarios, the cost of proving that a request for anonymity revocation is legitimate is not proportional to the compensation that the service provider gets.

A simple example is the following: to use a service, an anonymous user has to pay a small fee within 30 days. If the user, however, fails to do this, the service provider has to prove the non-payment towards the revocation authority in order to obtain the user's identity and take action. Distributing the revocation authority across multiple machines owned by different organisations does not solve this problem; on the contrary, now all of these organisations have to check non-payment, which further increases the costs for the service provider.

### Oblivious Satisfaction Authority

In scenarios similar to the aforementioned example, it is often easier for the user to prove satisfaction, than for the service provider to do the opposite. Therefore, in our new approach, we shift some responsibilities from the service provider to the user. Instead of the service provider having to prove to the revocation authority that the revocation conditions have been met, it is the user's responsibility to prove that the satisfaction conditions have been fulfilled. This change facilitates a far less complicated resolution of disputes and conflicts, which is both more economical for the service provider and more privacy preserving for the user.

The approach is as follows: upon the user's request, an Oblivious Satisfaction Authority (SA) verifies the satisfaction of a specific condition with respect to a specific service, and provides the user with a satisfaction token. The satisfaction authority can be made oblivious in the sense that the SA must not be able to link a user's satisfaction transaction with the user's transaction at the service provider. Moreover, even if the oblivious satisfaction authority and the oblivious revocation authority collude, they should not be able to link satisfaction requests with opening requests. This is achieved in a similar way as for the oblivious RA: the satisfaction token is in fact double encrypted, and the satisfaction authority is only able to remove the outer layer, while only the user is able to remove the final blinding.

After unblinding the satisfaction token received from SA, the user publishes this token, proving satisfaction towards the revocation authority. In other words, when the service provider requests the user's identity, he has to provide the same satisfaction token to the revocation authority. Now, the revocation authority only discloses the (blinded) identity to the service provider if the corresponding satisfaction token has *not* been published before some predefined date. If the user, however, decides not to fulfill the contract, and as such cannot publish the corresponding satisfaction tokens, the revocation authority discloses the user's identity to the service provider.

Since the satisfaction tokens can be machine verified, the involvement of the revocation authority can be reduced significantly and expensive external authorities such as law enforcement become obsolete. This combined approach with oblivious revocation and oblivious satisfaction authorities, better serves the needs of service providers, as it keeps the process of revocation and the dependency on external revocation authorities to a minimum. Furthermore, it provides better privacy guarantees towards the user than the solution with a fully trusted revocation authority.

To achieve this, the scheme restricts the revocation authority to only process blinded information, unknown to service providers, and to output blinded information that can only be decrypted by the user. As a result, SA cannot block requests of U selectively even when under pressure by the service provider, and it cannot collude against any specific user, nor can it link the transactions of users in the system.

These strong guarantees do not only protect the user, but they also simplify privacy friendly transactions. In particular, we can implement a form of anonymous payment based on credit cards rather than anonymous e-cash. When satisfying the payment condition towards the satisfaction authority, the user is identified (through



his credit card number); however, because of the unlinkability guarantee, his transaction with the service provider remains anonymous.

## 5.8 Conclusion

Even though a large number of very advanced privacy-enhancing cryptographic primitives have been proposed in the literature, their way to broad-scale deployment in the real world still presents a number of challenges.

One is the design of user interfaces that capture the core concepts of the underlying cryptography, while hiding the details.

Another challenge is the integration of the cryptographic primitives in the overall (authentication and access control) infrastructure. For instance, to deploy anonymous credentials, one needs proper policy languages to express and communicate the access control requirements in a way that supports, e.g., selective revealing of attributes, or proving properties of attributes. Too often do such languages implicitly assume that the user reveals all of her attributes by default. Moreover, since credential attributes are often sensitive information, these policy languages have to be integrated with privacy policy languages in which servers can express how the revealed information will be treated, and for users to express to whom and under which circumstances they are willing to reveal it. Privacy policy languages such as P3P [W3C06] are a first step, but are often not fine-grained enough, and lack the tight integration with access control policies. These and other challenges are currently being addressed as part of the PrimeLife project [pria, CMN<sup>+</sup>10].

From a cryptographic perspective, there are still many open problems to be addressed. Researchers are searching for more efficient primitives, since the incurred overhead is still prohibitive in many applications. Also, dedicated protocols for advanced applications like social networks or location-based services would be desirable. From a theoretical point of view, an important challenge is how existing primitives can be securely and efficiently composed to build new, more complex primitives. Finally, most of the above primitives currently still lack proper key management infrastructures so that keys can be securely stored, authenticated, and revoked.

# Chapter 6

## Transparency Tools

Hans Hedbom, Tobias Pulls, and Marit Hansen

**Abstract** The increasing spread of personal information on the Internet calls for new tools and paradigm to complement the concealment and protection paradigms. One such suggested paradigm is transparency and the associated transparency enhancing tools, making it possible for Data Subjects to track and examine how their data have been used, where it originates and what personal data about them that Data Controllers have stored. One such tool needed in order to track events related to personal data is a log system. Such a log system must be constructed in such a way that it does not introduce new privacy problems. This chapter describes such a log system that we call a privacy preserving secure log. It outlines the requirements for the system and describes and specifies a privacy preserving log system that has been developed and implemented within the PrimeLife project.

### 6.1 Introduction

As users of information technology, we are today giving away more and more personal information to many different actors. The personal information we expose can easily be transferred to third parties or harvested from public sources without the control (or consent) of the Data Subject. This situation, in combination with the fact that emerging technologies such as sensor networks and ambient intelligence environments, makes it even harder to control when or what information is collected, and has made several researchers argue that the paradigm of concealment and data minimization is no longer enough to protect or limit the exposure and release of personal data [Hil09, SSA06, WABL<sup>+</sup>06]. The solution that is suggested in [Hil09, SSA06, WABL<sup>+</sup>06] is to increase the transparency of how data is collected, stored and used in order to make it possible for the Data Subject to make informed decisions and to monitor the usage of her data and that agreed policies are honoured.

There are also legal provisions for transparency. In the European data protection framework, several provisions exist that support or demand transparency. Among

others, Art. 12 Data Protection Directive 95/46/EC specifies the legal provision that grants every person the right to access, i.e., the right to obtain from the controller, a confirmation as to whether data relating to him are being processed and information about the purposes of the processing, the categories of data concerned, and the recipients or categories of recipients to whom the data are disclosed. This is the basic provision for exercising other Data Subject rights such as rectification, erasure, or blocking of data (cf. Art. 12). Further, in any automatic processing of data concerning him, at least in the case of the automated decisions, Data Controllers should grant every Data Subject the knowledge of the logic involved. In addition to the Data Protection Directive 95/46/EC, other regulations may demand further information or notification processes. Outstanding is the personal data breach notification that is laid down in Art. 4 No 3 e-Privacy Directive 2009/136/EC: “In the case of a personal data breach, the provider of publicly available electronic communications services shall, without undue delay, notify the personal data breach to the competent national authority. When the personal data breach is likely to adversely affect the personal data or privacy of a subscriber or individual, the provider shall also notify the subscriber or individual of the breach without undue delay.” Summarising the mentioned provisions, the need for transparency for Data Subjects is apparent. This requires that the Data Controller has transparency of the data processing, which should be taken for granted, but is often not the case.

The technical and legal tools needed to achieve this type of transparency have been named “Transparency Enhancing Tools (TETs)” [Hil09]. There are some different definitions of what a TET is, however, our understanding of what constitutes a technical TET is the following, based on a definition in [Hed09] : A TET is a technical tool that accomplishes one or more of the following:

1. It provides information about the intended collection, storage and/or data processing to the Data Subject, or a proxy acting on behalf of the Data Subject, in order to enhance the Data Subject’s privacy;
2. It provides the Data Subject with an overview of what personal data have been disclosed to which Data Controller under which policies;
3. It provides the Data Subject, or her proxy, online access to her personal data, to information on how her data have been processed and whether this was in line with privacy laws and/or negotiated policies, and/or to the logic of data processing in order to enhance the Data Subject’s privacy;
4. It provides “counter profiling” capabilities to the Data Subject, or her proxy, helping her to “guess” how her data match relevant group profiles, which may affect her future opportunities or risks.

Within PrimeLife, several transparency tools have been developed covering points 1 and 2 in the list above and the data access part of point 3. The Data Track described in Chapter 3, for example, has all of these characteristics and the Privacy Dashboard has elements of 1 and 2 in it. However, since these tools are presented elsewhere in this book, we will concentrate on the processing part of point 3 in the definition in this section.

In the following, Section 6.2 will give a short example to set the scene. Section 6.3 will introduce the notion of Privacy Preserving Secure logging and outline and motivate the requirements of such a log system. Section 6.4 will describe previous work within the area, while Section 6.5 gives a technical overview of the log system. Finally, Section 6.6 concludes and gives a small outlook on future work.

## 6.2 Setting the Scene

So what good is transparency of processing for a user? Let us answer that question by giving a small example case using the Data Subject Alice. Assume in the following that Alice has an application capable of accessing and displaying in a user friendly manner the events that has happened to her data at different Data Controllers. This could, for example, be an added functionality of the Data Track.

Alice has an e-mail address that she is very restrictive in using for anything but personal communication with friends and family. Suddenly a lot of unsolicited e-mails (spam) is dropping into the mailbox of this address. Alice gets a bit annoyed and wonders what has happened. In order to try to find out, she fires up her Data Track and performs a search of released data giving her e-mail address as the search key. The search returns showing that she accidentally used this email address on two occasions: once when ordering a book at *bad books co* and once when she inquired about a flight at *Cheep Airlines Ltd.* Alice then asks the application to access these two services and retrieve the events that are related to her data. She then searches the events finding the ones that relates to the e-mail address. The result of this search turns up a record at *bad books co* showing that the e-mail address has been forwarded to five third party companies, all of which are the ones that have been sending unsolicited e-mails to her. Since the application also stores agreed policies, it informs her that this transfer is a violation of the agreement. Alice now knows that the address was leaked by *bad books co* ; this was a policy violation. With this information, she can take proper action against *bad books co* and ask for her data to be removed there and at the third party sites.

This is of course a limited and benign example, but similar actions could be taken to find out general policy violations, the reason for an automated decision or who and why personal data have been accessed. For more discussions on the benefits of transparency and examples see, e.g., [Hil09, SSA06, WABL<sup>+</sup>06].

The rest of this chapter will concentrate on one of the technologies behind the scene needed to implement process transparency in a privacy friendly manner, Privacy Preserving Secure logging.

### 6.3 On Privacy Preserving and Secure Logs

In order to know how personal data have been processed and used, there needs to be a way of keeping track of events related to the data in the system, e.g., when and how data have been accessed and by who, when or what process and what was the reason for that access. The most common way of keeping track of events in a computer system is to keep a number of logs storing events of a specific type or related to a specific aspect of the system, e.g., security logs store security related events and application logs store relevant events connected to a specific application. As a consequence, we argue that there needs to be detailed logging of how personal data have been processed and used by the data controller on behalf of the Data Subjects whose personal data are being processed. However, the privacy preserving secure log is supposed to be beneficial to the privacy of the Data Subjects whose data are being processed: it should not become a privacy problem in and of itself. Thus, this log needs to have special properties. In general, a log can be viewed as a record of sequential data. A secure log, when compared to a standard log, protects the confidentiality and integrity of the entries in the log. A privacy preserving secure log, in addition to being a secure log, tries to address the privacy problems related to the fact that you are keeping a log of how personal data are processed and used for the sake of transparency. Each entry in the privacy preserving secure log concerns one Data Subject, the entity on whose behalf the entry is made. The log is secure in the sense that confidentiality is provided by encrypting the data stored in entries and integrity is provided by using hashes and message authentication codes. The log is privacy preserving by providing the following properties:

1. The data in a log entry can only be read by the Data Subject to whom the entry relates. This ensures that no other entity can read the data stored in the log, which could violate the privacy of the data subject.
2. Data Subjects and the Data Controller can independently validate the integrity of the entries that concerns them. If multiple entities are needed to validate the integrity of parts of the log, no single entity will fully trust in the integrity of, and hence the data stored in, entries of the log.<sup>1</sup>
3. Unlinkability of Data Subject's entries; that is, you cannot tell to which Data Subject an entry relates. Without this property, it would be possible to tell how many entries there are in the log for each Data Subject, which might reveal, for example, how frequent a Data Subject is using a service.<sup>2</sup>
4. The Data Controller safely provides anonymous access to the log entries for Data Subjects. Requiring authenticated access to entries could allow an attacker to link entries to Data Subjects.

---

<sup>1</sup> The Data Controller can validate the integrity of all the entries in the log without knowledge of the contents.

<sup>2</sup> This is of course highly dependent on what is stored in the log.

### 6.3.1 Attacker Model and Security Evaluation

Our privacy preserving secure log provides the properties described previously for all entries committed to the log prior to an attacker compromising the Data Controllers system running the logging system. Once compromised, little can be done to secure or provide any privacy to future entries committed to the log. The attacker model is thus that the Data Controller is initially trusted and then at some point in time becomes compromised. Even if a large portion of the Data Subjects become compromised, we show in [HPHL10] that the properties of the log hold until the point in time when the Data Controller becomes compromised.

We present some arguments throughout the technical overview in section 6.5 as to why the properties in question hold for our log. A more complete security evaluation of our privacy preserving secure log can be found in [HPHL10]. For Data Subjects to learn when a Data Controller has been compromised, we assume that there exists some mechanism, such as software working with a TPM (such as the PRIME core) or regular audits by a trusted third party. Depending on the anonymity service used by Data Subjects to access the API of the log, there might be attacks that allow an attacker to link the downloaded entries to the Data Subject. One example would be the use of a low-latency anonymity network such as Tor and an attacker doing end-to-end correlation.<sup>3</sup>

## 6.4 Prior Work and Our Contribution

There exists a number of solutions on secure logging in the literature. The most relevant is the Schneier-Kelsey secure log [SK98], which was used as a foundation for our privacy preserving secure log, but also work done by Holt [Hol06], Ma et al. [MT07, MT08] and Accorsi [Acc08]. In general, these solutions provide confidentiality and integrity of log entries committed to the log prior to an attacker compromising the logging system. This is accomplished by depriving an attacker of access to one or more cryptographic keys used when creating the entries committed to the log, either by destroying old keys or using public key cryptography. Neither of the solutions mentioned fully addresses the problem of unlinkability and anonymous access. Some work of unlinkability in connection with logs has been addressed by Wouters et al. [WSLP08]. However, this work primarily addresses the unlinkability of logs between logging systems in an eGovernment setting rather than unlinkability of log entries within a log. Further, they do not address the problem of an inside attacker or provide anonymous access to log entries. Our main contributions are in the area of unlinkability and the ability to safely provide anonymous access to log entries. For details concerning our contributions and ongoing work, see [HPHL10, EP10, HH10]. Further work is being done on developing a distributed version of the log, allowing transparency logging to continue when data about a

---

<sup>3</sup> See <https://blog.torproject.org/blog/one-cell-enough>

Data Subject is shared from one Data Controller to another in a privacy-friendly manner.

## 6.5 Technical Overview

Conceptually, the privacy preserving secure log, hereafter referred to as simply “the log,” can be divided into four parts; state, entry structure, storage and the API. With the help of these four parts, we will explain how the properties outlined in the previous section are accomplished by the log.

### 6.5.1 State and Secrets

As stated earlier, each entry in the log relates to a Data Subject. When an entry is added to the log, both the data to be logged and the *identifier*<sup>4</sup> of the Data Subject for whom the entry is created has to be provided. To be able to log data for a Data Subject, the Data Subject’s identifier must have been initialised in the log’s *state*. Formally speaking, the state maps (entity, attribute name) to an attribute value.

When a new Data Subject is initialised in state, a unique identifier and three values, the Data Subject’s *secret*, *seed* and *public key*, are stored in the log’s state. These three values are provided by the Data Subject and only the Data Subject knows his private key. The secret and seed are large random values. For each Data Subject identifier, the state keeps track of the following attributes:

- **AK** - The authentication key for the next entry in the log. The initial value is derived from the Data Subject’s secret.
- **PK** - The public key.
- **ID** - The value of the identifier for the previous entry in the log for the Data Subject in an obfuscated form. The initial value, since for the first entry there is no previous entry, is the seed provided by the Data Subject.
- **Chain** - The value of the chain for the previous entry in the log for the Data Subject in an obfuscated form. There is no initial value for this attribute.

The log’s state evolves as new entries are added to the log. The old values stored in state, if overwritten as part of the state update procedure<sup>5</sup>, are irrevocably deleted from the Data Controller. This is accomplished by cryptographic hashes and message authentication codes, with the authentication key for the entry as part of either the data being hashed or as a key. This is the main procedure that leads to the “prior to” property described earlier; when the attacker compromises the Data Controller,

---

<sup>4</sup> The only requirement from the log is that all Data Subject identifiers are unique. A user could be known under any number of different data subject identifiers.

<sup>5</sup> Details can be found in “Adding Secure Transparency Logging to the Prime Core” [HPHL10].

the information needed to compromise the entries already committed to the log is missing or computationally hard to generate.

The Data Controller, like all Data Subjects, has an entry in the log's state with one additional attribute: a signing key used by the logging system for signing the data stored in entries. The initial secret and seed of the Data Controller's system should not be stored on the Data Controller to ensure that the "prior to" property holds true for the Data Controller as well.

### 6.5.2 Entry Structure and Storage

A log entry consists of five fields: two identifiers, two chains and the data field. The Data Controller and the Data Subject have an identifier and chain each, see [Figure 6.1](#).



Fig. 6.1: The five fields of an entry in the log.

The purposes of the different fields are:

- **The identifier field** contains a unique identifier that only the entity that the field belongs to can generate by having knowledge of the authentication key used to generate it. This is a hash.
- **The chain field** provides cumulative verification of the integrity of the entries in the log; either all entries in the log, for the Data Controller with help of the controller chain, or all the entries that belong to a specific Data Subject with help of the subject chain. This is the field that allows for independent integrity validation by each entity. This is a message authentication code.
- **The encrypted data field** provides data confidentiality by encrypting the data with the public key of the Data Subject, ensuring that only the Data Subject can read the data.

Log entries are stored in the log's storage. Storage is a multiset or bag, where entries are stored without any order. Entries can be retrieved from storage based on the value of the identifier field for the Data Controller or Data Subject.



### 6.5.3 API

The log provides an unauthenticated stateless API for Data Subjects to access entries stored in the log. It can be accessed anonymously by Data Subjects if they use an anonymising service such as Tor. The following two methods are exposed:

- **GetEntry(identifier)** - returns the entry with the given entry subject identifier from storage. If no entry is found, a dummy entry is generated and returned instead.
- **GetLatestEntryID(identifier)**<sup>6</sup> - returns the identifier stored in the log's state for the Data Subject with the given identifier.

Providing anonymous access to entries is safe because of how the entries are structured. An attacker lacking knowledge of the corresponding private key that decrypts an entry cannot read the entry's contents. An attacker without knowledge of the authentication key used to generate any identifier or chain cannot use the values to link any entries together.

### 6.5.4 Unlinkability

Unlinkability between log entries and Data Subjects for entries committed to the log prior to an attacker compromising the Data Controllers system is provided because:

- The data field is encrypted with the Data Subject's public key using a KEM-DEM [ISO] hybrid cipher using probabilistic encryption schemes with key-privacy[BBDP01]. This means that an attacker cannot learn which public key, out of all the public keys in the system, was used to encrypt the data by inspecting the encrypted data or by encrypting common log data and comparing the results.
- The identifier and chain fields are created using ideal hash and message authentication code algorithms respectively, where either as part of the data or as a key, an authentication key is used that is no longer known to an attacker.
- The entries are stored in a multiset, that is, they are not in chronological order. This is important to prevent correlation attacks, using other logs in the system such as the Apache access log. We address this issue for the implementation of our log in [HH10], where we present a solution that destroys the recorded chronological order of entries inserted into relational databases with minimal affect on average insert time for inserting entries into the log.

---

<sup>6</sup> This is a method that is not strictly needed: the Data Subject could query the GetEntry-method until an invalid entry is returned, but its use makes some attacks on the unlinkability property of the log harder for an attacker. Further, the returned value is encrypted using probabilistic encryption to prevent an attacker from determining when a new entry is added to the log for a Data Subject. See [EP10, HH10] for more information.

## 6.6 Conclusion and Outlook

Within PrimeLife, several tools that have the aspects of transparency enhancement have been developed, however, in this section we have discussed a tool to help the Data Subject (or an external auditor) determine how her personal data have been used by the Data Controller. A privacy preserving secure log enables a Data Controller to store log messages concerning how a Data Subject's personal data is processed without the log in and of itself becoming a privacy problem. We have discussed the requirements needed for such a log and by implementing it within the PrimeLife project shown that it is indeed possible to create a practical log system that fulfils all the requirements. The log itself is currently implemented as a module and can be integrated, fulfilling certain conditions, in a system under development. However, it is not that useful for a normal user unless it can be accessed easily and the information it contains can be shown in a user friendly manner. Because of this, we are now concentrating our efforts on providing a user interface that can present the log records in a way that is understandable for a user and that also automatically retrieves and verifies the log in a privacy-friendly manner. This user interface, when finished, could for example be integrated in the Data Track in order to create an application similar to the one Alice is using in the example. We would also like to investigate what events need to be logged in order to devise a method for establishing an optimal log strategy, trying to find a balance between completeness and efficiency.



## Chapter 7

# Interoperability of Trust and Reputation Tools

Sandra Steinbrecher and Stefan Schiffner

**Abstract** Reputation systems naturally collect information on who interacts with whom and how satisfied the interaction partners are about the outcome of the interactions. Opinion of and about natural persons are personal data and need to be protected. We elaborate requirements for electronic reputation systems. We focus on security properties, that is if a system is secure an attacker can not forge reputation values, further we elaborate privacy protection goals. A short literature survey on system implementations is given. We discuss interoperability of different reputation providers, that is how can a reputation be transported from one to an other reputation provider. Finally, we show how reputation systems should be integrated in identity management systems.

### 7.1 Introduction

Privacy enhancements constrain trust establishment in social software, since trust is usually based on information about the user's past behaviour observed by other users. Privacy enhancing technologies follow data minimisation and control paradigms, hence the access to user-related information is restricted by design. However, some information is needed to base trust upon. Crypto mechanisms, on one hand, allow for legal enforceability of (pseudonymous) interaction partners to prevent them from misbehaving. Many interactions, on the other hand, are informal, or it is too expensive to enforce liability. For these reasons, social software often deploys reputation systems as additional mechanisms for establishing trust in user interactions. Reputation objects are not only users, but products, web content and services; or more generally, anything users depend on for their goals. In Chapter 3 of this book we already outlined and discussed the scenario of web content. In this section, we generalise this scenario a more comprehensive one and outline the social needs (Section 7.2), the legal aspects (Section 7.3), the resulting requirements (Section 7.4),

and finally the technical implementability of reputation management (Section 7.5). We focus on its interoperability in (Section 7.6).

## 7.2 Social need

People maintain relations and rely on them in various ways. Who did not already book a hotel that a colleague has been to before? Who never read a book a friend recommended? Who does not sometimes go to the same doctor as his pals? When using the Internet, many users also rely on these relations and make use of Web 2.0 applications to make these links explicit. Those social networking software allows users to maintain a list of buddies/friends, share information with them and explore the social structure of their peer group.

With the growing usage of the Internet, the social network of humans covers the Internet as well. Hence, there is an interest of users in transferring the trust other human beings place in them in the offline world to the Internet. Also the respective applications are interested in getting information about their users from the offline world. In addition, people make links to other Internet users they did/do not know personally. These links can be made up explicitly by direct interaction, e.g., discussions in Internet communities; or the links might have made up implicitly by Internet applications that link users, e.g., according to common interests.

Users may not be totally aware of these links, e.g., when they are recommended books in an Internet shop. However, these links have a high potential to be used for trust decisions as in the offline world. In fact, from hotel ratings (e.g., TripAdvisor<sup>1</sup>) and book reviews (e.g., Amazon<sup>2</sup>) to medical advice, all physical world examples from above have their electronic counterpart and are present in many applications. Thus users desire to transfer the trust (the users of) one application place(s) in them also to other applications and their users. Also, the respective applications are interested in getting information about users from other applications to make up their own trust decision as to how to treat this user.

Social scientists and theoretical economists model the problem of whether two users who want to interact, should place trust in each other as trust games [CW88, Das00] that need inter-personal context-specific trust. In this context, the term social networks often refers to a modeling way of social individuals' relations. We represent those by a graph that has a vertex as representation for each individual and directed and labeled edges for relations. The links describe a possible relation the human beings have to each other, e.g., friendship, common interests, dislike.

A reputation network is a social network that links users (possibly pseudonymously) to each other and allows them to interact and exchange information with and about each other. For the concept of reputation, we assume that a user's past can predict his future behaviour. The creation of this prediction is called learning. Users

---

<sup>1</sup> <http://www.tripadvisor.com/> (last visited Jan. 2011)

<sup>2</sup> <http://www.amazon.com/> (last visited Jan. 2011)

within the reputation network can learn an object's reputation from other users in the network who share experiences with the reputation object. In social sciences, this is called the **learning mechanism** of the reputation network [BR01].

If users in the reputation network are reputation objects (e.g., as seller in a marketplace or author in a wiki) or can influence reputation objects (e.g., as author for the content created being the reputation object), users may control others in the reputation network by spreading information about the respective user. This will hopefully influence the user's behaviour positively. If the (behaviour of the) reputation object cannot be changed by the users it should at least be neglected by them. In social sciences, this is called the **control mechanism** of the reputation network [BR01].

Reputation systems assist reputation networks technically. To implement both the learning and the control mechanism of the reputation network, a reputation system has to offer the following functions to its users [Ste09]:

**Learning mechanism and evaluation function:** The reputation system provides users with an *evaluation function* to learn a reputation object's reputation following specific rules. Possibly, every evaluator might receive a different reputation of the reputation object. The selection of ratings used by the evaluation function depends on both the *information flow* of ratings in the reputation network and the *trust structure* in the reputation network, i.e., how users trust in others' ratings.

**Control mechanism with a rating function:** Control is exercised after an interaction takes place. It gives feedback to the user and these ratings are kept in a history for later evaluations. There are two types of users who can make use of the control mechanism: the interaction partner in the form of interaction-derived reputation and possible observers in the form of observed reputation [Mui03]. The reputation system provides authorised raters with a *rating function* that allows them to map reputation objects to ratings. The reputation system updates the reputation of the reputation object from the ratings received with a *reputation function*.

### 7.3 Legal Aspect

Reputation systems can be seen as databases that collect information about who interacted with whom in which context and the respective persons' opinion about each other based on this interaction. According to Bygrave [Byg02], opinions about a natural person are personal data, so that the respective person's right on informational self-determination is applicable. Therefore, explicit reputation should only be accumulated about users who agreed to accumulation. This reputation should only be shown to others after users give their consent. Furthermore, reputation information should be protected by means of technical data protection, as outlined by Mahler and Olsen [MO04]. The usage of pseudonyms can help here but if the granularity of the reputation is too fine, the reputation itself becomes a quasi-identifier that allows pseudonymous profile building.

## 7.4 Security and Privacy Requirements

As for many other technical systems, security and privacy requirements were not a major issue when the first reputation systems were designed and established. With their wider application, a growing number of reputation systems are subject to various attacks as outlined, e.g., in an ENISA position paper [ENI07]: Thus security and privacy requirements of reputation systems has been studied. This chapter is mainly based on [Ste09, ENI07, Vos04, SCS10]. However, these papers are written from different perspectives, while we focus on the technical functionalities of a reputation system as identified in Section 7.2. Those lead to the following building block requirements:

Rating function:

- *Accountability of ratings*: Users want raters to be accountable for the ratings they give for reputation objects.
- *Raters' anonymity*: Users want to rate anonymously.

Reputation function: The reputation system updates the reputation object's reputation from the ratings received. The rating function follows specific rules fixed by the system designer. These rules typically depend on the application scenario and have to fulfill sociological and economic requirements. However, the following requirements should hold:

- *Completeness of reputation*: Users want the aggregated reputation to consider all ratings given and that are available to him according to the information flow in the reputation network.
- *Liveliness of reputation*: Reputation should always consider all recent interactions or give users an indication that there are no more. Especially users who are reputation objects or can influence reputation objects should not have the possibility to reach a final state in which bad behaviour no longer damages the respective reputation.

Evaluation function: The aggregated reputation of a reputation object can be shown to other users on request. Therefore, the following requirements apply:

- *Availability of reputation*: All users in the reputation network need to be able to access a reputation object's reputation; however, if the reputation object is a user, this might require his consent from a legal perspective.
- *Evaluator's anonymity*: Users want to evaluate reputation anonymously to prevent others from building personal behaviour profiles of their interests.
- *Possibly reputation object's anonymity*: If the reputation object is a user, he might not want to be linked to his past interactions (except that these contributed to his reputation) to prevent others from building profiles about all his interactions and interaction partners.
- *Unlinkability of reputation objects*: If reputation objects have some relation to each other, this should not be revealed.

- *Persistence of reputation objects*: Users' reputations need to be persistent and enduring [RKZF00]. While the first property ensures that re-entering with a neutral reputation is hard, the latter ensures that a user has a long history to learn from.
- *Individual scalability of reputation*: Users want to be able to decide on the delivered reputation depending on the trust structure the reputation network has for them.

By actions in the reputation network, no other requirements on interactions in the reputation network should be affected. This calls for unlinkability of actions for the same user as well as for his anonymity when doing something.

## 7.5 Technical Implementability

Reputation systems have been widely studied in social sciences, economics and computer science. Special attention has been paid to the possible design of reputation system architectures and reputation functions, i.e how to calculate a reputation from given ratings. An overview of architectures is for example provided by Voss in [Vos04], while possible reputation functions are, for example, outlined by Mui in [Mui03]. For an economic introduction, we refer to Dellarocas' work [Del03].

It is quite clear that it is difficult or even impossible to design a reputation system that fulfills all security requirements. However, there exist a number of approaches that try to fulfill at least a significant subset. As the focus of our work is to outline how reputation systems can be applied to the scenario of privacy-enhanced social software and privacy-enhancing identity management, we concentrate on the following privacy requirements: anonymity of raters, evaluators, and reputation objects as well as unlinkability of reputation objects.

In [PRT04], using anonymity services to achieve privacy for reputation systems is proposed. However, this approach is inadequate since it only protects the evaluators. In order to obtain anonymity of raters and reputation objects, it needs to be ensured that many users are indistinguishable by an attacker, so that they are in large anonymity sets. For *unlinkability of reputation objects*, others should not be able to link interactions with the same user. The possibility of recognising users by reputation is limited if the set of possible reputations is limited [Ste09] or if the reputation is only published as an estimated reputation, as proposed in [Del00]. Transaction pseudonyms can be used to avoid linkability between transactions [ACBM08, Ste06]. In order to obtain *anonymity of raters*, interactions and ratings related to these interactions need to be unlinkable. This can be reached by a reputation provider who only calculates a new user reputation after it collected not only one but several ratings [Del06], or who only publishes an estimation of the actual reputation [Del00]. Further, a rater can be anonymous against the reputation provider by using convertible credentials [Ste09] or electronic cash [ACBM08, SCS10]. Furthermore, in [Ker09] a provable secure reputation system.



This system uses 2 TTP that ensure that ratings and reputations are unlinkable. However, it does not provide anonymity for the interaction partners, since the authors argue that this would be useless: interaction in the physical world requires addresses anyway for good delivery and money transfer. This is the reason why in [SCS11] a system based on DC-nets is proposed that provides *privacy* in the form of information theoretic relationship anonymity with regards to users.

It was a focus of Prime Life to contribute to the possible design options of privacy-respecting reputation systems as you can see from the publications already cited above [Ste09, SCS10, SCS11].

## 7.6 Infrastructure

For all functions of reputation systems, namely rating, reputation and evaluation function, infrastructure needs arise. This means, a reputation system either has to be integrated in other systems or be closely interoperable with them. In the following, we discuss interoperability for both options, as we already published as a result of PrimeLife in [Ste10].

### 7.6.1 Interoperability with Applications

Currently the vision arises to establish stand-alone reputation systems that collect information from various interactions in different applications.

According to the social needs of reputation systems outlined in Section 7.2, the applications, where the interactions rated took place, have to provide the reputation system with as much information as possible on the following aspects:

- *Model of Trust Game*: Only users who gave a leap of faith to reputation objects should be able to rate them. Applications have to make a clear model, who gave a leap of faith, and specify this for the reputation system.
- *Interaction information*: As reputation is context-dependent, information on the interaction rated is needed, e.g., time, value for the interaction partners.
- *Rater information*: As reputation needs to build on inter-personal trust, information on the raters is also needed, as will be outlined in Section 7.6.2

As there already exist a number of reputation systems integrated in applications, making these reputation system(s) interoperable becomes of interest. The problem of interoperability that is represented by the reputation exchange function in our model is twofold:

- *Format*: Firstly, formats for common exchange and possibly also internal representation of reputation are needed. An OASIS group<sup>3</sup> working on a possible

<sup>3</sup> [http://www.oasis-open.org/committees/tc\\_home.php?wg\\\_abbrev=orms](http://www.oasis-open.org/committees/tc_home.php?wg\_abbrev=orms) (last visited Jan. 2011)

portable format using XML. However, we still currently lack such a standard that could be implemented. Here is the need for solutions that can be easily integrated in the existing web technologies. The suggestion we implemented for PrimeLife in [SGM09] was to use the Resource Description Framework (RDF) common for the Web 2.0 and allowing for adding reputation information as meta-information to arbitrary web content.

- *Algorithm:* In every reputation system, different implementations of rating function, evaluation function and reputation function are defined depending on the system design. An algorithm for transferring reputations received from another reputation systems to the own reputation system is needed that balances possible advantages and disadvantages. This algorithm needs to comprise inheritance rules for reputations to decide on interoperability of reputation or ratings from different reputation systems.

The OpenPrivacy Initiative<sup>4</sup> presented Sierra, a reference implementation of a reputation management framework comprising several components representing the functions of the reputation system as well as an identity management system. They also define reputation exchange functions, whose actual implementation can be determined by the system designer in terms of exchange rates between reputations calculated from different reputation functions.

However, there are other issues of interoperability between reputation systems so far neglected by the technical literature:

- *Several reputation exchanges:* For several executions of the reputation exchange functions between two reputation systems, it has to be secured that it is clear which part of reputation has already been exchanged.
- *Related reputation objects:* So far, we assumed that the reputation object is well determined. Another issue of interoperability for reputation systems deals with is the possible relation between distinct reputation objects. An example is the wiki scenario from Chapter 1: The reputation system of collecting reputation of content might need to exchange reputation with the reputation system collecting reputation of authors. Certainly there is some relation between a content and its authors, but it might not be advisable to transfer reputation of one content directly to its authors and vice versa. Thus reputation systems need to define the transfer of reputation between related objects by a *reputation object exchange function*.

Aside from the OpenPrivacy Initiative, there are commercial stand-alone systems such as iKarma,<sup>5</sup> which is a “third-party service for collecting, managing and promoting [your] reputation among [your] customers and contacts” or portals like Trivago<sup>6</sup> that comprises reputation information from various other reputation systems.

The scientific approaches, that outline reputation infrastructures independent from concrete applications (e.g., [Vos04, PS08, KK09, SCS10, SCS11]), do not fol-

<sup>4</sup> <http://openprivacy.org/> (last visited Jan. 2011)

<sup>5</sup> <http://ikarma.com/support/faq/\#1> (last visited Jan. 2011)

<sup>6</sup> <http://www.trivago.com/> (last visited Jan. 2011)

low the centralised approach of the commercial solutions, but use local storage of reputation information to enable users to show the reputation they collected to others themselves. All of these suggestions need some external infrastructure to prevent reputation manipulation by the reputation object.

In the mentioned scientific approaches, the trust model is implicitly clear. However, as all of them aim for a privacy-respecting reputation system neither interaction nor rater information is provided. For the commercial solutions users can provide as much information as they want on themselves and their interactions.

### 7.6.2 Interoperability with Trust Management

As outlined in Section 7.2, reputation networks need to have some kind of inherent trust structure. When a user wants to determine a reputation object's credibility, that is trustworthiness, he has to determine his trust in two other sources as well:

- *Raters*: The ratings given by raters can be:
  - *subjective ratings*, that are influenced by the raters' subjective estimation of the reputation object, or
  - *objective ratings*, that can be verified by all other users than the rater at some point in time and that would have come to the same ratings.

An example for the first type of ratings is eBay while examples for the second type can be found in P2P systems, e.g., GNUnet<sup>7</sup>, where the reply to a query leads to a positive reputation, and a reply can be proved or verified at least at the time it is sent.

If the raters are humans, subjective ratings will be given. Then the rater needs to decide whether he would have come to the same rating; this means their views on the reputation object is interoperable. For this reason, a trust management system to determine the inter-personal trust in raters is needed. It can be realised by an additional reputation system for raters.

- *Reputation systems*: Evaluators need to have system trust in all reputation systems that collected the ratings and calculated the reputation the user evaluates.

Technically, trust management is often associated with PKI structures [Mau96] (beneath other approaches). PKI structures allow for binding keys to pseudonyms. Others can use their key to sign this binding. Thereby chains to other users, who want to trust in this binding, can be built. These chains can be constructed hierarchically with certification authorities or in the form of a web of trust (e.g., GPG/PGP). Both structures can also be used for the broader deployment of reputation systems. Hierarchies and chains as they work for trust management can be applied to reputation management to express which experiences from others can be trusted.

However, the straightforward approach to implement ratings as signatures and use existing PKI structures only assures accountability of keys and linkage to their

<sup>7</sup> [www.gnunet.org](http://www.gnunet.org) (last visited April 2010)

holder. However, if a user or certification authority signs someones key in a PKI structure, that does not say anything about the credibility/competence they assume the key holders to have as reputation objects. For this reason, different key(s) than for accountability are needed and existing certificate structures have to be extended appropriately.

### ***7.6.3 Interoperability with Identity Management***

For the evaluation function of reputation systems, not only the overall reputation, but also the single ratings and the raters who gave them, might be important. If raters misbehave maliciously by giving ratings that do not reflect the concrete experience they had with reputation objects, there should be a possibility to detect this and probably to make them accountable for it.

As for the collection of large reputation profiles about users (both reputation objects and raters), privacy becomes an important issue. A reputation system should be interoperable with privacy-enhancing user-controlled identity management systems (PE-IMS). An IMS in general is able to certify users and grant rights to them for applications. Additionally a PE-IMS [CPHH02, CK01] like PRIME<sup>8</sup> assists users platform-independently in controlling their personal data in various applications and selecting pseudonyms appropriately, depending on their wish for pseudonymity and unlinkability of actions.

The interoperability of a reputation system with a PE-IMS needs a privacy-respecting design of reputation systems, while keeping the level of trust provided by the use of reputations as outlined in [Ste09].

When a reputation system interoperates with a PE-IMS, it is possible and intended that users have several partial identities (pIDs) which cannot be linked, neither by other users using the systems nor by the underlying system. Hence, both raters and reputation objects are only known by pseudonyms to each other.

If there would exist only one reputation value per user, all pIDs of this user would have the same reputation. This would ease the linking of the pIDs of one user because of the same reputation value. Thus, having separated reputations per pID and not only one per user is a fundamental condition for a reputation system in the context of identity management.

The use of pIDs raises the problem that a malicious user may rate himself a lot of times using new self-created pID for every rating in order to improve his own reputation. This kind of attack is also known as a Sybil attack [Dou02]. If the reputation system is not defined carefully, it would be easy for such an attacker to improve their own reputation unwarranted. This can be limited/prevented by entrance fees or the use of once-in-a-lifetime credentials as suggested in [FR99]. When using PRIME as IMS, the latter can be implemented by its identity provider issuing such credentials. Alternatively or additionally, fees could also be collected.

---

<sup>8</sup> Privacy and Identity Management for Europe (<http://www.prime-project.eu/>), funded by the European Union in the 6. Framework Program, 2004-2008.

### 7.6.4 Resulting implementation

For users as reputation objects, we outline in the following paragraphs a possible resulting implementation. Our design description is independent from concrete rating, reputation and evaluation functions.

All communication is secured by encryption to reach confidentiality of all ratings and actions performed. Also all messages can be transferred in an anonymous way with an anonymous communication network. All actions and ratings are secured by digital signatures (given under a pseudonym using PRIME) for integrity reasons.

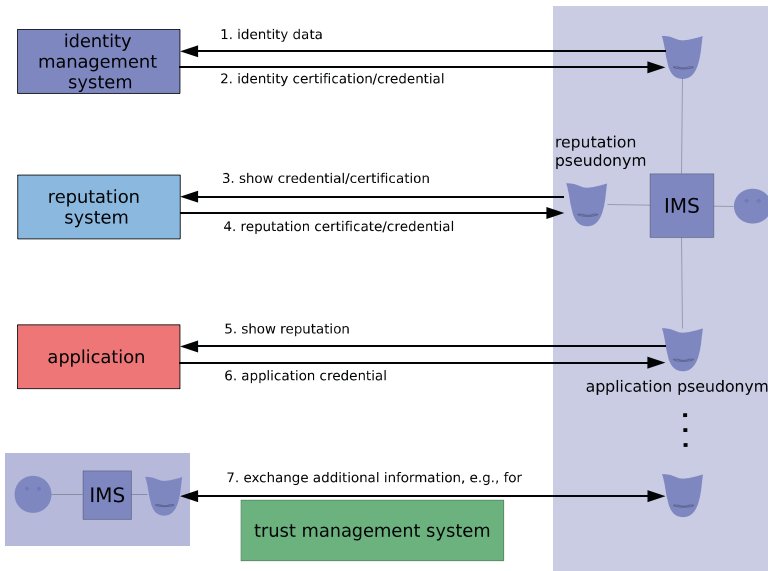


Fig. 7.1: Infrastructure for users as reputation objects [Ste10].

For identity management, a user registers himself with an identity management system (provider) by declaration of his identity data (Step 1 in Fig. 7.1). After verifying the data, the identity provider issues a credential or certification on (part of) these data (Step 2 in Fig. 7.1). By the use of an identity management system (provider), accountability of the pseudonym can be given.

When the user wants to register with a reputation system (provider), he sends the reputation system the certification/credential he got from the identity management system (provider) (Step 3 in Fig. 7.1). This should guarantee that no user is able to build up a reputation under multiple pseudonyms within the same context and every user can be identified in the case of misbehaviour. The reputation system (provider)

creates a reputation certificate/credential based on the certificate/credential from the identity management system (provider) and sends it back to the user (Step 4 in Fig. 7.1).

The reputation credential contains the user's reputation pseudonym, his initial reputation and possibly other attributes such as the applications it can be used in or an expiration date.

Based on the reputation credential, the user can register himself with an application by showing his reputation certificate/credential (Step 5 in Fig. 7.1). He thereby agrees that he will collect a reputation for his interactions within the application (e.g., a marketplace or a wiki) with the reputation system he registered with. Based on this, he gets an application credential to use the application (Step 6 in Fig. 7.1).

Additionally, the user might interact with other users to exchange additional information, e.g., via a trust management system to inform himself about this user (possibly as a rater) and other users in the reputation network (Step 7 in Fig. 7.1).

Every action the user performs above can be done under distinct pseudonyms if convertible credentials are issued by the respective providers.

We implemented this infrastructure for phpBB as application and the user-controlled privacy-enhancing identity management PRIME as outlined in [PS08]. Currently we lack a trust management in our implementation.

## 7.7 Conclusion

Although many proposals for sophisticated reputation systems exist in scientific literature, many applications on the Internet (especially the ones usable free of charge) just test simple reputation systems and adapt them slightly based on the experience the providers make. Especially security and privacy issues of users and providers get more and more important nowadays to prevent certain threats and attacks. Within PrimeLife, it was possible to contribute to this trend by making suggestions on interoperability and security aspects and implementing first tools that show how these issues addressed.



## Chapter 8

# Data Privacy

Michele Bezzi, Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, Stefano Paraboschi, and Pierangela Samarati

**Abstract** In today's globally interconnected society, a huge amount of data about individuals is collected, processed, and disseminated. Data collections often contain sensitive personally identifiable information that need to be adequately protected against improper disclosure. In this chapter, we describe novel information-theoretical privacy metrics, necessary to measure the privacy degree guaranteed by a published dataset. We then illustrate privacy protection techniques, based on fragmentation, that can be used to protect sensitive data and sensitive associations among them.

### 8.1 Introduction

In the modern digital society, information is one of the most valuable and demanded resources. Nowadays, organisations and end users are resorting to service providers for disseminating and sharing the huge amount of collected data they want to make available to others. Although this solution guarantees high data availability at a reduced price, it introduces new privacy and security concerns. Indeed, the collected datasets often include sensitive personally identifiable information, which are no longer under the direct control of their owners. In such a scenario, guaranteeing the privacy of the data, be them published or outsourced to a service provider, becomes a primary requirement.

A first step in the definition of methods that guarantee privacy protection in public or semi-public release consists in the definition of privacy metrics, measuring the degree of protection offered by a published dataset. Recently, most of this line of work has focused on  $k$ -anonymity [Sam01] and its variations (e.g.,  $\ell$ -diversity [MGK06] and  $t$ -closeness [LLV07]), which guarantee that the released dataset satisfies a given protection degree, represented by the value of  $k$  ( $\ell$  and  $t$ , resp.). These approaches basically define a minimal requirement (worst-case scenarios) that each combination of entries in the dataset should satisfy. Although  $k$ -anonymity and its variations



are simple and effective, the privacy degree they offer can be neither combined in a unified privacy metric nor compared. To overcome these issues, novel metrics have recently been proposed [Bez10] that express the disclosure risk in terms of information theory. These solutions are based on the concept of one-symbol information, which determines the contribution of each entry to the risk of disclosure and allows for assessing the privacy offered by privacy protection techniques. The modeling of the privacy problem, in the line of research mentioned above, typically assumes a setting where data to be protected are either quasi-identifiers or sensitive information associated with them. Novel proposals have instead considered the more general problem of protecting arbitrary sensitive associations among data [ABG<sup>+</sup>05]. These solutions are based on fragmenting data to break associations among them that should not be disclosed. The use of fragmentation for satisfying privacy needs has first been proposed in the data outsourcing scenario, where data are stored and managed by an external service provider, to improve query evaluation efficiency. Indeed, traditional approaches assume that an overlying layer of encryption is applied on the data before outsourcing them (e.g., [HIM02a]). Based on the observation that often what is sensitive is the association among data, more than the data per se, novel proposals resort to fragmentation, possibly combined with encryption, for privacy protection [ABG<sup>+</sup>05, CDF<sup>+</sup>07, CDF<sup>+</sup>09b].

In this chapter, we illustrate novel privacy metrics for assessing privacy protection techniques and we describe recent proposals, based on the use of fragmentation, for protecting data privacy. The remainder of this chapter is organised as follows. Section 8.2 describes privacy metrics based on information theory concepts. Section 8.3 introduces the basic concepts on which fragmentation-based proposals for protecting the privacy of data rely. Section 8.4 presents an approach combining fragmentation and encryption to protect sensitive associations among data. Section 8.5 illustrates a proposal that departs from encryption and where a small portion of the data is stored on the data owner's side to break sensitive associations. Section 8.6 describes how fragmentation can also be adopted in the data publication scenario, possibly complementing fragments with loose associations, representing in a sanitised form the associations broken by fragmentation, to increase data utility. Finally, Section 8.7 concludes the chapter.

## 8.2 Privacy Metrics and Information Theory

In the literature, several models have been proposed for capturing different aspects of the disclosure risk [FWCY10]. From the data publisher's point of view, it would be desirable to have these privacy models expressed in terms of semantically "similar" measures, so she could be able to compare their impact and optimise the trade-off between the different privacy risks. In [AA01], the authors proposed an information theoretic framework to express average disclosure risk using mutual information. The advantages of mutual information formulation are twofold: *i)* it allows for expressing the different risk measures, and associated thresholds, in

Original dataset				Anonymised dataset			
SSN	DoB	ZIP	Illness	SSN	DoB	ZIP	Illness
123456	56/02/04	26010	Measles	**	195*/**/**	26***	Measles
234561	58/11/07	26015	Asthma	**	195*/**/**	26***	Asthma
345271	52/09/07	26123	Flu	**	195*/**/**	26***	Flu
456291	71/06/07	40765	Flu	**	197*/**/**	40***	Flu
563810	78/05/14	40123	H1N1	**	197*/**/**	40***	H1N1
678451	78/05/02	40672	Flu	**	197*/**/**	40***	Flu
782340	81/97/11	70128	Gastritis	**	198*/**/**	70***	Gastritis
895641	85/01/01	70542	Chest pain	**	198*/**/**	70***	Chest pain

(a)

(b)

Fig. 8.1: An example of a dataset (a) and its anonymised version (b).

a common framework, with well defined units; *ii*) it permits the application of a wide range of well established information theory tools to risk optimisation (e.g., privacy-distortion trade-off problem [RMFDF09]). In this section, we will present some recent results on the information theoretic formulation of privacy risk measures.

8.2.1 Basic Concepts

Existing privacy metrics ( $k$ -anonymity [Sam01],  $\ell$ -diversity [MGK06] and  $t$ -closeness [LLV07]) define minimal requirements for each entry in the dataset, but because mutual information is an average quantity, frameworks expressing average disclosure risk using mutual information are not able to completely express these conditions on single entries. In fact, as pointed out in [LL09], *privacy is an individual concept and should be measured separately for each individual*. Accordingly, average measures such as mutual information are not able to fully capture privacy risk. To overcome this limitation, we should consider one-symbol information (i.e., the contribution to mutual information by a single record), and define the disclosure risk metrics accordingly [Bez10]. By introducing one-symbol information, it becomes possible to express and compare different risk concepts, such as  $k$ -anonymity,  $\ell$ -diversity and  $t$ -closeness, using the same units. In addition, it is possible to obtain a set of constraints on the mutual and one-symbol information for satisfying  $\ell$ -diversity and  $t$ -closeness, and also to determine a relationship between the risk parameters  $t$  and  $\ell$ , which allows us to assess  $t$  in terms of the more intuitive  $\ell$  value.

Traditionally, all sensitive data that need to be protected are stored in a unique relation  $r$  over relation schema  $R(a_1, \dots, a_n)$ , with  $a_i$  an attribute on domain  $D_i, i = 1, \dots, n$ . From a disclosure perspective, attributes in  $R$  can be classified as follows:

- *Identifiers*. Attributes that uniquely identify respondents (e.g., SSN).

- *Quasi-identifiers (QIs)*. Attributes that, in combination, can be linked to external information to re-identify all or some of the respondents, or reduce the uncertainty over their identities (e.g., `DateOfBirth`, `ZIP`, `Gender`).
- *Sensitive attributes*. Attributes that contain sensitive information about respondents (e.g., `Illness`, `Salary`, `PoliticalParty`).

There are two types of disclosure: identity disclosure, and attribute disclosure. Identity disclosure occurs when the identity of an individual can be reconstructed and associated with a record in the released dataset. Attribute disclosure occurs when an attribute value can be associated with an individual (without necessarily being able to link to a specific record). In anonymising the original data, we want to prevent both kinds of disclosure. In the anonymisation process, identifiers are suppressed (or replaced with random values), but this is not sufficient, since by combining the quasi-identifier values with some external source information (e.g., a public register), an attacker could still be able to re-identify a subset of the records in the dataset.

Let us consider a dataset containing identifiers, quasi-identifiers (referred to as  $X$ ), and sensitive attributes, (referred to as  $W$ ). Figure 8.1(a) illustrates an example of a relation including an identifier =  $\{SSN\}$ , a quasi-identifier  $X = \{DOB, ZIP\}$ , and a sensitive attribute  $W = \{Illness\}$ . We create an anonymised version of such data, removing identifiers, and anonymising the quasi-identifier ( $\tilde{X}$ ), for example substituting its original values with more general ones [Sam01]. Figure 8.1(b) reports an example of a dataset obtained by anonymising the relation in Figure 8.1(a).

## 8.2.2 Traditional Privacy Metrics

Among the metrics proposed so far, let us describe three of them, which well illustrate the different aspects of privacy risks.

$k$ -anonymity [Sam01] condition requires that *every* combination of quasi-identifier attributes (QI group) is shared by at least  $k$  records in the anonymised dataset. A large  $k$  value indicates that the anonymised dataset has a low identity disclosure risk, because, at best, an attacker has a probability  $1/k$  to re-identify a record, but it does not necessarily protect against attribute disclosure. In fact, a QI group (with minimal size of  $k$  records) could also have the same value for the sensitive attribute, so even if the attacker is not able to re-identify the record, she can discover the sensitive information.

$\ell$ -diversity [MGK06] captures this kind of risk.  $\ell$ -diversity condition requires that, for *every* combination of quasi-identifier attributes, there should be at least  $\ell$  “well represented” values for each sensitive attribute. In [MGK06], a number of definitions of “well represented” were proposed. Because we are interested in describing an information-theoretic framework, the more relevant definition for us is in terms of entropy,

$$H(W|\tilde{x}) \equiv - \sum_{w \in W} p(w|\tilde{x}) \log_2 p(w|\tilde{x}) \geq \log_2 \ell$$

for every QI group  $\tilde{x}$ , and with  $\ell \geq 1$ ,  $p(\tilde{x})$  is the frequency of the QI group  $\tilde{x}$  and  $p(w|\tilde{x})$  is the relative frequency of the sensitive attribute  $w$  for a given QI group  $\tilde{x}$ . For example, if each QI group has  $n$  equally distributed values for the sensitive attributes, the entire dataset will be  $n$ -diverse. Note that if the  $\ell$ -diversity condition holds, the  $k$ -anonymity condition (with  $k \leq \ell$ ) also automatically holds, since there should be at least  $\ell$  records for each group of QIs.

Although the  $\ell$ -diversity condition prevents a possible attacker from inferring the exact sensitive attribute values, she may still learn a considerable amount of probabilistic information. In particular, if the distribution of sensitive attributes within a QI group is very dissimilar from the distribution over the whole set, an attacker may increase his knowledge on sensitive attributes (*skewness attack*, see [LLV07] for details).  $t$ -closeness [LLV07] estimates this risk by computing the distance between the distribution of confidential attributes within the QI group and in the entire dataset. The authors in [LLV07] proposed two ways to measure the distance, one of them has a straightforward relationship with mutual information, as described in the following.

### 8.2.3 An Information Theoretic Approach for Privacy Metrics

Following [Bez10], we can reformulate the privacy metrics illustrated in Section 8.2.2 in terms of information metrics, therefore expressed using the same units (bits).

#### $k$ -anonymity

In case of suppression and generalisation, a single QI group in the anonymised dataset  $\tilde{x}$  can correspond to a number,  $N_{\tilde{x}}$  of records in the original table. Accordingly, the probability of re-identifying a record  $x$  given  $\tilde{x}$  is simply:  $p(x|\tilde{x}) = 1/N_{\tilde{x}}$ , and  $k$ -anonymity reads:

$$H(X|\tilde{x}) \geq \log_2 k \quad (8.1)$$

for each  $\tilde{x} \in \tilde{X}$ . In terms of *one-symbol specific information* [DM99]  $I_2 \equiv I_2(X, \tilde{x}) \equiv H(X) - H(X|\tilde{x})$ , where  $H(X|\tilde{x}) \equiv - \sum_{\tilde{x} \in \tilde{X}} p(\tilde{x}|x) \log_2 p(\tilde{x}|x)$ , it reads

$$I_2(X, \tilde{x}) \equiv H(X) - H(X|\tilde{x}) \leq \log_2 \frac{N}{k} \quad (8.2)$$

where  $N$  is the number of tuples in the original dataset  $X$  (assumed different).  $I_2(X, \tilde{x})$  measures the identity disclosure risk for a single record. Equation 8.1 holds

also in case of perturbative masking [Bez07], therefore  $I_2$  can be used for any kind of masking transformations.

Averaging Eq. 8.2 over  $\tilde{X}$  we get:

$$I(X, \tilde{X}) \leq \log_2 \frac{N}{k}$$

So, the mutual information can be used as a risk indicator for identity disclosure [DFRM09], but we have to stress that this condition does not guarantee the  $k$ -anonymity for every QI group  $\tilde{x}$ , that is, it is a necessary, although not sufficient, condition.

### **$t$ -closeness**

$t$ -closeness condition requires:

$$D(p(w|\tilde{x})||p(w)) \equiv \sum_{w \in W} p(w|\tilde{x}) \log_2 \frac{p(w|\tilde{x})}{p(w)} \leq t \quad (8.3)$$

for each  $\tilde{x} \in \tilde{X}$ . This is equivalent to the one-symbol information  $I_1$  (surprise) [Fan61], that is:

$$I_1(W, \tilde{x}) \equiv \sum_{w \in W} p(w|\tilde{x}) \log_2 \frac{p(w|\tilde{x})}{p(w)} \leq t \quad (8.4)$$

$I_1(W, \tilde{x})$  is a measure of attribute disclosure risk for a QI group  $\tilde{x}$ , as the difference between the prior belief about  $W$  from the knowledge of the entire distribution  $p(w)$ , and the posterior belief  $p(w|\tilde{x})$  after having observed  $\tilde{x}$  and the corresponding sensitive attributes. Averaging over the set  $\tilde{X}$ , we get an estimation of the disclosure risk (based on  $t$ -closeness) for the whole set [RMFDF09], as follows:

$$I(W, \tilde{X}) \equiv \sum_{\tilde{x} \in \tilde{X}} p(\tilde{x}) \sum_{w \in W} p(w|\tilde{x}) \log_2 \frac{p(w|\tilde{x})}{p(w)} \leq t \quad (8.5)$$

Again, this is a necessary but not sufficient condition to satisfy  $t$ -closeness on an entire table, since this condition requires to satisfy  $t$ -closeness for each  $\tilde{x}$ .

### **$\ell$ -diversity**

$\ell$ -diversity condition, in terms of entropy, reads:

$$H(W|\tilde{x}) \geq \log_2 \ell$$

for each QI group  $\tilde{x} \in \tilde{X}$ . It can be expressed in terms of one-symbol specific information  $I_2$ , as follows:

$$I_2(W, \tilde{x}) \equiv H(W) - H(W|\tilde{x}) \leq H(W) - \log_2 \ell \quad (8.6)$$

$I_2(W, \tilde{x})$  is a measure of attribute disclosure risk for a QI group  $\tilde{x}$ , as the reduction of uncertainty between the prior distribution and the conditional distribution. Averaging over the set  $\tilde{X}$ , we get an estimation of the average disclosure risk for the whole set [RMFDF09].

$$I(W, \tilde{X}) \equiv H(W) - H(W|\tilde{X}) \leq H(W) - \log_2 \ell \quad (8.7)$$

This is the  $\ell$ -diversity condition on average. Again, this is a necessary but not sufficient condition to satisfy  $\ell$ -diversity for each  $\tilde{x}$ . Note that the mutual information is a non-negative quantity ( $I(W, \tilde{X}) \geq 0$ ). From Equation 8.7 immediately follows that  $H(W)$  is an upper bound for  $\log_2 \ell$ , that is,  $\log_2 \ell \leq H(W)$  or equivalently  $\ell \leq \ell_{\max} \equiv 2^{H(W)}$ .

### Comparing Risk Parameters

Equations 8.5 and 8.7 suggest a way to compare the two risk parameters  $\ell$  and  $t$ . Indeed, if we equalise the maximal contribution to information of  $\ell$  and  $t$ , we can derive the following relation:

$$\ell_t = 2^{H(W)-t} \quad (8.8)$$

$\ell_t$  tells us, for a given  $t$ , what the *equivalent* value  $\ell$  is, that is, the value of  $\ell$  that has the same impact on the information. The advantage of Equation 8.8 is that it allows us to express the value of  $t$  parameter, which is often hard to set, in terms of  $\ell$  that has a more intuitive meaning.

In summary, for any anonymised dataset that satisfies  $\ell$ -diversity and  $t$ -closeness, the following average conditions are *necessary*:

$$\begin{cases} I(W, \tilde{X}) \leq I(W, X) \\ I(W, \tilde{X}) \leq t \\ I(W, \tilde{X}) \leq H(W) - \log_2 \ell \end{cases} \quad (8.9)$$

whereas the *necessary* and *sufficient* conditions are:

$$\begin{cases} I_1(W, \tilde{x}) \leq t \\ I_2(W, \tilde{x}) \leq H(W) - \log_2 \ell \end{cases} \quad (8.10)$$

for each  $\tilde{x} \in \tilde{X}$ .

For setting the risk parameters, lower and upper bounds for  $\ell$  are:

$$1 \leq \ell \leq \ell_{\max} \equiv 2^{H(W)} \quad (8.11)$$

and the  $\ell_t$  equivalent to  $t$ , is:

$$\ell_t = 2^{H(W)-t}$$

that allows for expressing  $t$  in terms of the more intuitive diversity parameter  $\ell$ .

In [LLV07], proposing  $t$ -closeness, the authors stated that “*Intuitively, privacy is measured by the information gain of an observer.*” The question is which metric we should use for measuring such information gain. In [Bez10], the author showed that if we consider information gain as a reduction of uncertainty, the corresponding privacy metric is similar to  $\ell$ -diversity, whereas if we think of information gain as the *novelty* of the information,  $t$ -closeness is the corresponding metric. Accordingly, the choice of the privacy risk metric depends on what kind of information we do not want to disclose, which in turn depends on the specific application, the tolerable level of information loss, and the attack model. The advantage of the formulation in terms of information theory is that it allows for expressing all the different metrics using comparable units (bits), and, at least in principle, use all the tools of information theory for optimising the privacy constraints, and, possibly, utility.

### 8.2.4 Protecting Privacy of Sensitive Value Distributions

The privacy metrics described in the previous section have been designed to measure the risk of identity and attribute disclosure in microdata release. However, similar metrics can also be used for different scenarios. For instance, in [BDLS10], the authors propose a model to assess the exposure of sensitive information in a scenario of data release, based on the concept of one-symbol information. The scenario concerns the incremental release of tuples of a table  $r$  defined over relation schema  $R(a_1, \dots, a_n)$ . Requested tuples, singularly considered, are not sensitive; however, their aggregation may allow for the inference of sensitive information not explicitly represented in the released dataset. In particular, inference is caused by peculiar value distributions of some released data. As an example, the age distribution of military personnel posted in a given location can allow an observer to deduce the sensitive nature of the location itself (e.g., a training campus or a headquarter). The authors propose a metric able to assess the disclosure risk caused by the release of a tuple.

The model defines sensitive entities in  $r$  as *sensitive properties* of targets, that are values of one or more non-sensitive released attributes  $Y \subset R$ . A sensitive property of target  $y \in D_Y$ , where  $D_Y$  is the domain of  $Y$ , is referred to as  $s(y)$ . For instance, considering the relation schema  $R(\text{Name}, \text{Age}, \text{Location})$  where `Name` is the name of a soldier, `Age` is her age, and `Location` is the location where the soldier is posted, targets can be the values of attribute `Location` (e.g.,  $L_1, \dots, L_n$ ) and the sensitive property of each target can be the related location type (e.g.,  $s(L_1) = \text{Headquarter}, \dots, s(L_n) = \text{TrainingCampus}$ ). Inferences on the sensitive property  $s(y)$  of target  $y$  can be caused by the distribution of values of some other attributes  $X \subseteq R$  for the specific  $y$  (i.e.,  $P(X|y)$ ), when there is a dependency between attributes  $X$  and  $Y$ . For instance,  $X = \{\text{Age}\}$ : in fact,  $P(\text{Age}|L_i)$  can reveal the location type (sensitive property) of target  $L_i$ . Dependency between attributes  $X$  and  $Y$  introduces an inference channel in a dataset concerning such attributes. In particu-

lar, inferences on the values of  $s(y)$  arise when  $P(X|y)$  shows peculiarities, differing from a typical (*baseline*) distribution of  $X$  that is expected and publicly known. The data model proposes the definition of  $X$ -outliers, targets for which the difference between  $P(X|y)$  and the baseline exceeds a given threshold.  $X$ -outliers show unusual distributions of  $P(X|y)$ , and this peculiarity induces in the observer an information gain that exposes the value of sensitive property  $s(y)$  [BDLS10]. To assess such information gain, the authors recur to  $I_1(y, X)$ , and compare it to the average value represented by the mutual information  $I(X, Y)$ . In this way, the authors can evaluate when the contribution of a particular target to the mutual information is unusually large, exposing as a consequence the value of the related sensitive property.

We note, however, that since observers do not have access to the original relation (supposed to be stored in a trusted environment), they can only see and learn distributions from the released collection of data: for this reason, the definition of actual  $X$ -outliers of the original relation may not fit what is observable. Therefore, the authors suggest that the evaluation of single target contributions to the mutual information be enforced runtime, on the released dataset, after a sufficient amount of tuples has been released to make the observable distributions representative of the original relation content. Therefore, in [BDLS10], the authors propose the evaluation of one-symbol information  $I_1(y, X)$  on the released dataset as a means for quantitatively assessing the risk of the release of a requested tuple.

## 8.3 Privacy Protection Techniques

Different protection techniques have been proposed to guarantee the privacy of the data when they are stored and managed by a third party (i.e., a service provider). Since the service provider is not under the direct control of the data owner, the privacy of the data may be put at risk. Traditional solutions (e.g., [DFPS07]) for protecting privacy of outsourced data assume that an overlying layer of encryption is applied on data before outsourcing. These solutions, however, highly reduce query evaluation efficiency. Also, they are based on the assumption that outsourced data are all equally sensitive, and therefore encryption is a price to be paid for their protection. However, in many scenarios, what is sensitive is the association among data, more than data items per se. In the following sections of this chapter, we will describe novel privacy protection techniques that, based on this observation, limit the use of encryption (or avoid it completely) by complementing it with fragmentation.

### 8.3.1 Basic Concepts

Privacy requirements can be conveniently modeled through *confidentiality constraints*. A confidentiality constraint  $c$  over relation schema  $R(a_1, \dots, a_n)$  is a subset of attributes of  $R$ , i.e.,  $c \subseteq R$ . The semantics of a confidentiality constraint  $c$  are that,



MEDICALDATA						
SSN	Name	DoB	ZIP	Illness	Treatment	
123456789	Alice	84/07/31	26015	Pharyngitis	Antibiotic	$c_0 = \{SSN\}$
231546586	Bob	82/05/20	26010	Flu	Aspirin	$c_1 = \{Name, DoB\}$
378565241	Carol	20/01/30	50010	Gastritis	Antacid	$c_2 = \{Name, ZIP\}$
489754278	David	80/07/02	20015	Broken Leg	Physiotherapy	$c_3 = \{Name, Illness\}$
589076542	Emma	75/02/07	26018	Gastritis	None	$c_4 = \{Name, Treatment\}$
675445372	Fred	75/02/17	26013	Asthma	Bronchodilator	$c_5 = \{DoB, ZIP, Illness\}$
719283746	Gregory	70/05/04	26020	Diabetes	Insulin	$c_6 = \{DoB, ZIP, Treatment\}$
812345098	Henrik	65/12/08	20010	Cancer	Chemotherapy	

(a)

(b)

Fig. 8.2: An example of relation (a) and of a set of well-defined constraints over it (b).

for each tuple in  $r$ , the joint visibility of the values of the attributes in  $c$  is sensitive, and needs to be protected. In particular, depending on the number of attributes involved, confidentiality constraints can be classified as follows.

- *Singleton constraints.* A singleton constraint states that the *values* of the attribute involved in the constraint are sensitive and cannot be released. For instance, the SSN of patients of a given hospital must be protected from disclosure.
- *Association constraints.* An association constraint states that the *association* among the values of the attributes in the constraint is sensitive and cannot be released. Even if, singularly taken, the values of the attributes in the constraint are not sensitive, the joint visibility of their values must be prevented. For instance, the association between the Name and the Illness of a patient has to be protected from disclosure.

The definition of confidentiality constraints, while immediate, provides great flexibility in the characterisation of the desired protection requirements. We note that the satisfaction of a constraint  $c_i$  implies the satisfaction of any constraint  $c_j$  such that  $c_i \subseteq c_j$ . Therefore, a set  $\mathcal{C} = \{c_1, \dots, c_m\}$  of confidentiality constraints is said to be *well-defined* if and only if  $\forall c_i, c_j \in \mathcal{C}, j \neq i, c_i \not\subseteq c_j$ .

*Example 8.1.* Figure 8.2 illustrates an example of relation (MEDICALDATA) along with a set of well defined confidentiality constraints, modeling the following privacy requirements:

- the list of SSNs of patients is considered sensitive ( $c_0$ );
- the association of patients' names with any other information in the relation is considered sensitive ( $c_1, \dots, c_4$ );
- attributes DoB and ZIP can work as a quasi-identifier [Sam01] and therefore can be exploited to infer the identity of patients; as a consequence, their associations with both Illness and Treatment are considered sensitive ( $c_5$  and  $c_6$ ).

Given a relation  $r$  defined over relation schema  $R(a_1, \dots, a_n)$ , the set  $\mathcal{C}$  of confidentiality constraints defined over  $R$  identifies the privacy requirements that must

be enforced when outsourcing  $r$ . Most of the proposals in the literature have put forward the idea of combining *fragmentation* and *encryption* techniques to satisfy confidentiality constraints. Both encryption and fragmentation, consistently with the formulation of confidentiality constraints, operate at the attribute level, that is, they involve attributes in their entirety. In particular, these techniques work as follows.

- *Encryption*. The values assumed by the attribute are encrypted, tuple by tuple, to make them visible only to authorised users.
- *Fragmentation*. The attributes in  $R$  are partitioned in different subsets, referred to as fragments, that are outsourced instead of  $R$ , to make attributes in different fragments jointly visible only to authorised users. A fragment  $F_i$  of a relation  $R$  is a subset of the attributes in  $R$  (i.e.,  $F_i \subseteq R$ ), and a fragmentation  $\mathcal{F}$  is a set of fragments over  $R$  (i.e.,  $\mathcal{F} = \{F_1, \dots, F_m\}$ ). A fragment instance of  $F_i$  is the set of tuples obtained by projecting the tuples in  $r$  over the attributes composing  $F_i$ .

Fragmentation and encryption can be combined in different ways to enforce confidentiality constraints. In particular, proposals in the literature differ on how original relation schema  $R$  is fragmented and whether encryption is applied. The first strategy proposed in the literature to enforce confidentiality constraints through fragmentation and encryption has been illustrated in [ABG<sup>+</sup>05]. This strategy is based on the assumption that data can be stored on two (or more) *non-communicating servers*. The relation schema  $R$  is therefore partitioned in two (or more) different fragments, each stored on a different server. The confidentiality constraints that cannot be enforced by means of fragmentation are solved by encrypting at least one of the attributes in the constraint. Since the assumption of the presence of two (or more) non-communicating servers is limiting in practice (collusion among servers can compromise the protection of sensitive data), new techniques have recently been proposed. These solutions nicely couple fragmentation and encryption to possibly store all the fragments in  $\mathcal{F}$  on a unique server. In Sections 8.4 and 8.5, we will describe these techniques in more details.

## 8.4 Fragmentation and Encryption

The proposal illustrated in [CDF<sup>+</sup>07], and refined in [CDF<sup>+</sup>09a, CDF<sup>+</sup>10], suggests the combined use of fragmentation and encryption techniques to enforce confidentiality constraints, while removing the need for storing fragments on non-communicating servers. In this section, we illustrate the data fragmentation model proposed in [CDF<sup>+</sup>07], we briefly describe how to compute a fragmentation enforcing a set of confidentiality constraints, and we finally illustrate the query evaluation process over fragmented data.

### 8.4.1 Fragmentation Model

The technique proposed in [CDF<sup>+</sup>07] enforces confidentiality constraints through the combined use of fragmentation and encryption. Intuitively, singleton constraints can only be enforced by means of encryption. Association constraints can be enforced via either fragmentation, storing attributes in the constraint in different fragments, or encryption, encrypting at least one of the attributes in the constraint. An advantage of this approach is that fragments can all be stored on the same server, since only authorised users, who know the encryption key, can join them. A fragmentation  $\mathcal{F} = \{F_1, \dots, F_m\}$  is *correct* if and only if the following two conditions hold: i)  $\forall c \in \mathcal{C}, \forall F \in \mathcal{F}, c \not\subseteq F$ ; and ii)  $\forall F_i, F_j \in \mathcal{F}, i \neq j, F_i \cap F_j = \emptyset$ . The first condition states that a fragment cannot contain in the clear all the attributes composing a confidentiality constraint (constraints satisfaction). The second condition states that fragments must be disjoint, since otherwise common attributes could be exploited to join fragments and reconstruct sensitive associations.

At a physical level, each fragment  $F_i \in \mathcal{F}$  is translated into a *physical fragment* that contains all the attributes in  $F_i$  in the clear, and all the other attributes in  $R$  are encrypted. Reporting all the attributes of  $R$  in each fragment, in either encrypted or plain form, permits the execution of any query on a single physical fragment, thus making the query evaluation process more efficient. The schema of the physical fragment  $F_i^e$  storing fragment  $F_i$  is  $F_i^e(\underline{salt}, enc, a_{i_1}, \dots, a_{i_m})$ , where:

- $\underline{salt}$  is the primary key of  $F_i^e$  and contains a randomly chosen value;
- $enc$  represents the encryption of all the attributes in  $R$  that do not belong to  $F_i$ , combined before encryption via a binary XOR with a salt, to prevent frequency-based attacks;
- $a_{i_1}, \dots, a_{i_m}$  are the attributes in fragment  $F_i$ .

Since the availability of plaintext attributes in a fragment makes query evaluation more efficient, fragmentation should be preferred to encryption in constraints satisfaction. To minimise the adoption of encryption, and *maximise visibility*, each attribute not appearing in a singleton constraint must belong to at least one fragment. As a consequence, a fragmentation that is correct and maximises visibility requires that each attribute not involved in a singleton constraint must appear in plaintext in exactly one fragment.

*Example 8.2.* Consider relation MEDICALDATA in Figure 8.2a and the set of constraints over it in Figure 8.2b. An example of a correct fragmentation that maximises visibility is  $\mathcal{F} = \{\{Name\}, \{DoB, ZIP\}, \{Illness, Treatment\}\}$ . Figure 8.3 illustrates the fragment instances over the physical fragments corresponding to  $\mathcal{F}$ . Note that only the attribute SSN does not appear in the clear in the fragments, since it belongs to singleton constraint  $c_0$ .

$F_1^e$			$F_2^e$				$F_3^e$			
salt	enc	Name	salt	enc	DoB	ZIP	salt	enc	Illness	Treatment
$s_1^1$	$\alpha$	Alice	$s_1^2$	$\vartheta$	84/07/31	26015	$s_1^3$	$\pi$	Pharyngitis	Antibiotic
$s_2^1$	$\beta$	Bob	$s_2^2$	$\iota$	82/05/20	26010	$s_2^3$	$\rho$	Flu	Aspirin
$s_3^1$	$\gamma$	Carol	$s_3^2$	$\kappa$	20/01/30	50010	$s_3^3$	$\sigma$	Gastritis	Antacid
$s_4^1$	$\delta$	David	$s_4^2$	$\lambda$	80/07/02	20015	$s_4^3$	$\tau$	Broken Leg	Physiotherapy
$s_5^1$	$\varepsilon$	Emma	$s_5^2$	$\mu$	75/02/07	26018	$s_5^3$	$\upsilon$	Gastritis	None
$s_6^1$	$\zeta$	Fred	$s_6^2$	$\nu$	75/02/17	26013	$s_6^3$	$\phi$	Asthma	Bronchodilator
$s_7^1$	$\eta$	Gregory	$s_7^2$	$\xi$	70/05/04	26020	$s_7^3$	$\chi$	Diabetes	Insulin
$s_8^1$	$\theta$	Henrik	$s_7^2$	$o$	65/12/08	20010	$s_7^3$	$\psi$	Cancer	Chemoterapy

Fig. 8.3: An example of a correct fragmentation in the multiple fragments scenario.

### 8.4.2 Minimal Fragmentation

Given a relation schema  $R$  and a set  $\mathcal{C}$  of confidentiality constraints over it, there may exist different correct fragmentations that maximise visibility. As an example, a fragmentation  $\mathcal{F}$  with a singleton fragment for each attribute  $a \in R$  that does not belong to a singleton constraint is correct and maximises visibility. However, this solution highly impacts the efficiency of the evaluation of queries involving more than one attribute. It is therefore necessary to identify a fragmentation that *i)* is correct, *ii)* maximises visibility, and *iii)* maximises efficiency in query evaluation.

Several criteria have been proposed for designing an optimal fragmentation, aiming at reducing the cost of query evaluation. A simple approach consists in minimising the number of fragments in  $\mathcal{F}$ . The rationale to this metric is that reducing the number of fragments implies that more attributes are stored in the clear in the same fragment, thus improving query execution efficiency. The problem of computing a fragmentation with the minimum number of fragments is however NP-hard (the hypergraph colouring problem [GJ79] reduces it). It is therefore necessary to determine a heuristic algorithm able to compute a good, even if not optimal, fragmentation. To this purpose, in [CDF<sup>+</sup>07], the authors introduce the concept of *dominance* between two fragmentations. A fragmentation  $\mathcal{F}'$  dominates a fragmentation  $\mathcal{F}$ , denoted  $\mathcal{F} \preceq \mathcal{F}'$ , if  $\mathcal{F}'$  can be obtained merging two (or more) fragments in  $\mathcal{F}$ . The dominance relation clearly states that a fragmentation  $\mathcal{F}$  is more convenient than a fragmentation  $\mathcal{F}'$  such that  $\mathcal{F} \preceq \mathcal{F}'$ . The heuristic proposed in [CDF<sup>+</sup>07] is therefore based on the computation of a *minimal fragmentation* with respect to the dominance relationship  $\preceq$ . A minimal fragmentation is a fragmentation  $\mathcal{F}$  that fulfills the following three requirements: *i)*  $\mathcal{F}$  is correct; *ii)*  $\mathcal{F}$  maximises visibility; and *iii)* there is not another fragmentation  $\mathcal{F}'$  that is correct, that maximises visibility and such that  $\mathcal{F} \preceq \mathcal{F}'$ .

To provide a more precise measure of the quality of a fragmentation, alternative solutions have been proposed. In line with traditional fragmentation solutions, the quality of a fragmentation can be measured in terms of the affinity between the attributes represented in the clear in the same fragment [CDF<sup>+</sup>10]. Since attribute affinity permits only the measurement of the advantage of storing in the same frag-

ments pairs of attributes, in [CDF<sup>+</sup>09a], the authors proposed a more precise approach, based on the estimation of query evaluation costs, considering the expected query workload on  $\mathcal{F}$ .

We note that both the problems of maximising affinity and minimising query evaluation costs when computing a correct fragmentation are NP-hard. Following this observation, ad-hoc heuristics have been proposed, tailored to the specific fragmentation metric [CDF<sup>+</sup>09a, CDF<sup>+</sup>10]. These algorithms exploit the fact that both affinity [CDF<sup>+</sup>10] and cost functions [CDF<sup>+</sup>09a] are monotonic with respect to the dominance relationship  $\preceq$ . This means that increasing the number of attributes represented in the clear in the same fragment, the quality of the fragmentation also increases.

### 8.4.3 Query Evaluation

A clear effect of the fragmentation process is that queries formulated by users on relation schema  $R$  must be evaluated on physical fragments, instead of on the original relation. As introduced in Section 8.4.1, each physical fragment stored in place of relation schema  $R$  contains, in either encrypted or clear form, all the attributes in  $R$ . As a consequence, each query can be evaluated by accessing one physical fragment only (the one offering better performances). Let us consider, for simplicity, a query  $q$  operating on relation  $R$ , of the form  $\text{SELECT } A \text{ FROM } R \text{ WHERE } Cond$ , where  $A$  is a subset of attributes in  $R$ , and  $Cond = \bigwedge_i cond_i$  is a conjunction of basic predicates of the form  $(a_i \text{ op } v)$ ,  $(a_i \text{ op } a_j)$ , or  $(a_i \text{ IN } \{v_i, \dots, v_k\})$ , where  $a, a_i, a_j \in R$ ,  $\{v, v_1, \dots, v_k\}$  are constant values in the domain of attribute  $a_i$ , and  $op$  is a comparison operator in  $\{=, \neq, >, <, \geq, \leq\}$ . For simplicity, in the following, we will use notation  $Attr(cond)$  to represent the attributes on which basic condition  $cond$  is defined.

Given the fragment  $F$  on which the query must be evaluated, first the condition in the WHERE clause of the query is split into two sub-conditions, depending on the party (server or client) that can evaluate it, as follows:

- $Cond_s = \bigwedge_i cond_i : Attr(cond_i) \subseteq F$  is the conjunction of conditions involving attributes included in the chosen fragment that, being reported in plaintext, can be evaluated by the server;
- $Cond_c = \bigwedge_i cond_i : Attr(cond_i) \subseteq (R \setminus F)$  is the conjunction of basic conditions involving attributes not included in the fragment that, being encrypted, can be evaluated only by the client.

After condition  $Cond$  has been split, the original query is translated into two queries:  $q_s$  operating on the server side, and  $q_c$  operating at the client side on the result  $R_s$  of  $q_s$ . The query executed by the server operates on the selected physical fragment  $F^e$ , evaluates condition  $Cond_s$ , and returns *salt*, *enc* and the attributes in  $F$  that appear in the SELECT clause of  $q$ . When the client receives the result  $R_s$  of query  $q_s$ , it decrypts attribute *enc* and evaluates, on the resulting tuples, condition

Original query	Translated queries
$q :=$ SELECT Treatment FROM MedicalData WHERE Illness='Asthma' AND ZIP='26013'	$q_s :=$ SELECT salt, enc, Treatment FROM $F_3^e$ WHERE Illness='Asthma'
	$q_c :=$ SELECT Treatment FROM Decrypt( $R_s$ .enc,salt,k) WHERE ZIP='26013'

Fig. 8.4: An example of query translation in the multiple fragments scenario.

$Cond_c$ . Finally, the client projects the result on the attributes in  $A$ . Note that if  $Cond_c$  is empty and  $A_s = A$ ,  $q_c$  does not need to be executed and  $q_s$  does not need to return attributes  $salt$  and  $enc$ , since  $R_s$  coincides with the result of the original query  $q$ .

*Example 8.3.* Consider relation MEDICALDATA in Figure 8.2a, the set of constraints over it in Figure 8.2b, and the fragmentation in Figure 8.3. Suppose now that a client formulates a query  $q =$  SELECT Treatment FROM MedicalData WHERE Illness='Asthma' AND ZIP='26013' returning the treatments to asthma adopted in zone 26013. The query can be translated to operate on any of the three fragments, but the evaluation using  $F_3^e$  is more convenient than using  $F_1^e$  or  $F_2^e$ , since  $F_3^e$  contains a subset of the attributes on which  $q$  formulates its conditions. The translation of query  $q$  in the corresponding queries operating on the server and on the client side, using  $F_3^e$ , is illustrated in Figure 8.4.

### 8.5 Departing from Encryption

The combined use of fragmentation and encryption presents several advantages, since the use of encryption is limited to the enforcement of singleton constraints. Although cryptographic tools enjoy today limited computational complexity, encryption carries the burden of key management and of expensive query evaluation over encrypted data. To overcome these limitations, in [CDF<sup>+</sup>09b], the authors put forward the idea of completely departing from encryption for constraint satisfaction. This solution is based on the assumption that the data owner is willing to store a limited portion of the data to enforce confidentiality constraints. In this section, we illustrate the data fragmentation model proposed in [CDF<sup>+</sup>09b], we briefly describe how to compute a fragmentation that enforces confidentiality constraints while minimising the data owner's workload, and we finally discuss how queries can be evaluated on fragmented data.

### 8.5.1 Fragmentation Model

The proposal illustrated in [CDF<sup>+</sup>09b] enforces confidentiality constraints by storing a subset of the data at the owner side, while the remaining information is outsourced to a service provider. Intuitively, confidentiality constraints are enforced by storing at least one attribute for each constraint on the data owner's side. In this scenario, the fragmentation process produces a pair  $\mathcal{F} = \langle F_o, F_s \rangle$  of fragments, where  $F_o$  is stored on the data owner, and  $F_s$  is stored on the external storage server. A fragmentation  $\mathcal{F} = \langle F_o, F_s \rangle$  is *correct* if and only if it satisfies the following two conditions: i)  $\forall c \in \mathcal{C}, c \not\subseteq F_s$ ; and ii)  $F_o \cup F_s = R$ . The first condition states that fragment  $F_s$  does not violate any confidentiality constraint. Note that fragment  $F_o$  does not need to satisfy this condition, since it is stored on the data owner's side and is therefore accessible only to authorised users. The second condition states that all the attributes in  $R$  must be represented either on the data owner or on the external server, to prevent loss of information. Although the data owner is willing to store a portion of the data, her storage capacity is limited. As a consequence, attributes that belong to  $F_s$  should not be replicated in  $F_o$  as well, since redundancy is not required. In other words,  $F_o$  and  $F_s$  should be disjointed, to avoid the replication of attributes already stored on the server side also on the data owner's side. At the physical level, however, the two fragments must have a common key attribute necessary to reconstruct the content of the original relation  $r$  (lossless join property). This attribute can be either the primary key of relation  $R$ , if it is not sensitive, or an attribute that does not belong to the schema of the original relation  $R$  and that is added to both fragments after the fragmentation process. At a physical level, a fragmentation  $\mathcal{F} = \langle F_o, F_s \rangle$ , where  $F_o = \{a_{o_1}, \dots, a_{o_i}\}$  and  $F_s = \{a_{s_1}, \dots, a_{s_j}\}$ , is translated into physical fragments  $F_o^e(\underline{\text{tid}}, a_{o_1}, \dots, a_{o_i})$  and  $F_s^e(\underline{\text{tid}}, a_{s_1}, \dots, a_{s_j})$ , where  $\text{tid}$  is the common tuple identifier.

*Example 8.4.* Consider relation MEDICALDATA in Figure 8.2a and the set of well defined constraints over it in Figure 8.2b. An example of a correct and non-redundant fragmentation is  $F_o = \{\text{SSN}, \text{Name}, \text{ZIP}\}$  and  $F_s = \{\text{DoB}, \text{Illness}, \text{Treatment}\}$ . Figure 8.5 illustrates the fragment instances over the physical fragments corresponding to  $F_o$  and  $F_s$ . Constraint  $c_0$  is satisfied by storing attribute SSN in  $F_o$ . Constraints  $c_1, \dots, c_4$  are satisfied by storing attribute Name in  $F_o$ . Constraints  $c_5$  and  $c_6$  are satisfied by storing attribute ZIP in  $F_o$ .

### 8.5.2 Minimal Fragmentation

Given a relation schema  $R$  and a set  $\mathcal{C}$  of confidentiality constraints over it, there may exist different fragmentations that are both correct and non-redundant. For instance, consider a fragmentation  $\mathcal{F} = \langle F_o, F_s \rangle$  where  $F_o = R$  and  $F_s = \emptyset$ . This fragmentation is clearly correct and non-redundant, but it is not desirable since it corresponds to no outsourcing. Among all the correct and non-redundant fragmentations,

$F_o^e$				$F_s^e$			
tid	SSN	Name	ZIP	tid	DoB	Illness	Treatment
1	123456789	Alice	26015	1	84/07/31	Pharyngitis	Antibiotic
2	231546586	Bob	26010	2	82/05/20	Flu	Aspirin
3	378565241	Carol	50010	3	20/01/30	Gastritis	Antacid
4	489754278	David	20015	4	80/07/02	Broken Leg	Physiotherapy
5	589076542	Emma	26018	5	75/02/07	Gastritis	None
6	675445372	Fred	26013	6	75/02/17	Asthma	Bronchodilator
7	719283746	Gregory	26020	7	70/05/04	Diabetes	Insulin
8	812345098	Henrik	20010	8	65/12/08	Cancer	Chemoterapy

Fig. 8.5: An example of a correct fragmentation in the departing-from-encryption scenario.

it is necessary to compute a solution that minimises the data owner's workload in terms of storage and/or intervention in the query evaluation process.

To compute a fragmentation that minimises the data owner's workload, it is first necessary to define a metric that quantitatively measures the cost of a fragmentation. This metric can be defined in different ways, depending on which resource is considered more valuable by the data owner (whose consumption should be minimised), and on the information available about the system workload at fragmentation time. Furthermore, there are different ways for measuring the consumption of the same resource and for combining metrics that analyse the use of different resources. As an example, in [CDF<sup>+</sup>09b], the authors propose four metrics, each corresponding to a different minimisation problem:

- *Min-Attr*: minimises the number of attributes in  $F_o$ ;
- *Min-Size*: minimises the physical size of the attributes in  $F_o$ ;
- *Min-Query*: minimises the number of queries that involve at least one attribute in  $F_o$ ;
- *Min-Cond*: minimises the number of conditions in queries that are evaluated over attributes in  $F_o$ .

We note that the minimisation of the number of attributes implies a minimisation of the storage space used at the data owner side, as well as a minimisation of the number of queries that require an involvement of the data owner. However, if more precise information about attributes' size or the query workload is available, it is possible to adopt more precise metrics in the computation of a fragmentation that minimises storage (*Min-Size*) or computation and bandwidth (*Min-Query* and *Min-Cond*) resources.

Independently from the fragmentation metric adopted, the problem of computing a minimal fragmentation is NP-hard (as demonstrated in [CDF<sup>+</sup>09b], the minimum hitting set problem reduces to it). As a consequence, in [CDF<sup>+</sup>09b], the authors propose a heuristic algorithm able to compute a minimal fragmentation in polynomial time, with respect to the number of attributes in  $R$ . The main advantage of this heuristic is its flexibility, since it can be adopted with any metric.



Server-Owner strategy	Owner-Server strategy
$q_s := \text{SELECT tid}$ FROM $F_s^e$ WHERE Illness='Asthma'	$q_o := \text{SELECT tid}$ FROM $F_o^e$ WHERE ZIP='26013'
$q_{so} := \text{SELECT Name}$ FROM $F_o^e$ JOIN $R_s$ ON $F_o^e.tid=R_s.tid$ WHERE ZIP='26013'	$q_s := \text{SELECT tid}$ FROM $F_s^e$ WHERE Illness='Asthma' AND tid IN {6}
	$q_{so} := \text{SELECT Name}$ FROM $F_o^e$ JOIN $R_s$ ON $F_o^e.tid=R_s.tid$

Fig. 8.6: An example of query translation in the departing-from-encryption scenario.

### 8.5.3 Query Evaluation

Since the attributes composing  $R$  are partitioned in two fragments to satisfy confidentiality constraints, the queries formulated by users on  $R$  must be translated into an equivalent set of queries on  $F_o^e$  and  $F_s^e$ . To this purpose, condition  $Cond = \bigwedge_i cnd_i$  in the WHERE clause of the original query  $q$  must be split in sub-conditions, depending on the party in the system who can evaluate it. More precisely,  $Cond$  is split in the following three conditions:

- $Cond_o = \bigwedge_i cnd_i : Attr(cnd_i) \subseteq F_o$  is the conjunction of basic conditions that can be evaluated only by the data owner;
- $Cond_s = \bigwedge_i cnd_i : Attr(cnd_i) \subseteq F_s$  is the conjunction of basic conditions that can be evaluated by the server, since they involve only attributes stored at the server;
- $Cond_{so} = \bigwedge_i cnd_i : Attr(cnd_i) \cap F_o \neq \emptyset$  and  $Attr(cnd_i) \cap F_s \neq \emptyset$  is the conjunction of basic conditions of the form  $(a_i \text{ op } a_j)$ , where  $a_i \in F_o$  and  $a_j \in F_s$ , that can be evaluated only by the data owner, with the support of the server.

The evaluation of a query  $q$  on  $R$  can follow two different strategies, depending on the order in which conditions  $Cond_s$ ,  $Cond_o$ , and  $Cond_{so}$  are evaluated, as described in the following.

- *Server-Owner strategy* first evaluates condition  $Cond_s$  on the server side and then evaluates both  $Cond_o$  and  $Cond_{so}$  on the data owner side.
- *Owner-Server strategy* first evaluates condition  $Cond_o$  on the data owner side, then evaluates condition  $Cond_s$  on the server side, and finally refines the result evaluating  $Cond_{so}$  on the data owner side.

*Example 8.5.* Consider relation MEDICALDATA in Figure 8.2a, the set of constraints over it in Figure 8.2b, the fragmentation in Figure 8.5, and query  $q = \text{SELECT Name FROM MedicalData WHERE Illness='Asthma' AND ZIP='26013'}$  returning the name of patients suffering from asthma in zone 26013. Condition

$Cond$  in the WHERE clause of  $q$  is split as follows:  $Cond_s=(\text{Illness}=\text{'Asthma'})$ ,  $Cond_o=(\text{ZIP}=\text{'26013'})$ , and  $Cond_{so}=\emptyset$ . Figure 8.6 illustrates how  $q$  is translated into an equivalent set of queries, following both the *Server-Owner* and the *Owner-Server* strategies.

The choice between the *Server-Owner* and the *Owner-Server* strategies depends on the possible leakage of information that the *Owner-Server* strategy may cause, as well as on their efficiency. In fact, if query  $q$  is not secret, the server knows the identifiers of the tuples satisfying condition  $Cond_o$ , possibly reconstructing sensitive associations. For instance, consider the example in Figure 8.6, the server knows that the query with  $\text{tid}=6$  is the unique tuple in  $R$  where  $\text{ZIP}=\text{'26013'}$ , thus violating constraints  $c_5$  and  $c_6$ .

## 8.6 Preserving Utility in Data Publication

Although fragmentation has been proposed as a method for the enforcement of confidentiality constraints in the data outsourcing scenario, it can also be adopted in data publishing, since it permits the publishing of views (fragments) only on data that do not expose sensitive associations. To increase the utility of published information, the fragmentation process should take into consideration the recipients' needs for information and the purpose of data publication. The fragmentation process can therefore be driven by *visibility requirements*, expressing views over data that the fragmentation should satisfy. Furthermore, fragments can be complemented with the release of the sensitive associations broken by fragmentation in a sanitised form as *loose associations*, defined in a way to guarantee a specified degree of privacy. In this section, we describe the proposal in [DFJ<sup>+</sup>10a] for guaranteeing utility in data publication by both defining a fragmentation that satisfies visibility requirements and publishing loose associations.

### 8.6.1 Visibility Requirements

Visibility requirements model views over the data that the fragmentation process should guarantee. A visibility requirement  $v$  over a relation schema  $R(a_1, \dots, a_n)$  is a monotonic boolean formula over  $\{a_1, \dots, a_n\}$ . The semantics of a visibility requirement  $v$  are that the visibility over the view represented by  $v$  should be guaranteed by the fragmentation. We note that the negation operator cannot be used in the definition of visibility requirements, since it corresponds to requiring the non-visibility over values, which is already captured by confidentiality constraints. Visibility requirements permit the expression of different needs of visibility, as described in the following.

- *Attributes visibility.* The values of the attribute in the visibility requirement should be released ( $v = a$ ).
- *Association visibility.* The association among the values of the attributes (views, in general) in the visibility requirement should be released ( $v = a_i \wedge \dots \wedge a_j$ ).
- *Alternative views.* At least one among the views composing the visibility requirement should be released ( $v = v_i \vee v_j$ ).

A fragmentation  $\mathcal{F}$  satisfies a set  $\mathcal{V}$  of visibility requirements if, for each requirement  $v \in \mathcal{V}$ , there exists at least a fragment  $F \in \mathcal{F}$  that satisfies  $v$ . We note that all the visibility requirements must be satisfied, but not necessarily by the same fragment in  $\mathcal{F}$ . A fragmentation is therefore *correct* and can be published only if it satisfies both confidentiality constraints and visibility requirements.

*Example 8.6.* Consider relation MEDICALDATA in Figure 8.2a. An example of a set  $\mathcal{V}$  of visibility requirements is represented by:

- $v_1 = \text{SSN} \vee \text{Name}$  states that either the SSN or the Name of patients should be released;
- $v_2 = \text{Illness} \wedge \text{Treatment}$  states that the association between Illnesses and Treatments should be released;
- $v_3 = \text{Name} \vee (\text{DoB} \wedge \text{ZIP})$  states that either the Name of patients, or the DoB and ZIP of patients in association should be released.

Fragmentation  $\mathcal{F} = \{\{\text{Name}\}, \{\text{DoB}, \text{ZIP}\}, \{\text{Illness}, \text{Treatment}\}\}$  in Figure 8.3 satisfies the visibility requirements in  $\mathcal{V}$ :  $v_1$  is satisfied by  $F_1$ ,  $v_2$  is satisfied by  $F_3$ , and  $v_3$  is satisfied by  $F_1$ . Fragmentation  $\mathcal{F}$  also satisfies the confidentiality constraints in Figure 8.2b. As a consequence,  $\mathcal{F}$  represents a correct fragmentation.

### 8.6.2 Loose Associations

Even if fragments in a correct fragmentation cannot be joined, in [DFJ<sup>+</sup>10a], the authors propose publishing *loose associations* among their tuples. A loose association reveals some information on the association broken by fragmentation, while guaranteeing that a given privacy degree is respected (to guarantee that confidentiality constraints cannot be violated). Intuitively, loose associations hide tuples participating in the associations in *groups* and provide information on the associations only at the group level (in contrast to the tuple level).

Let us consider a pair  $F_l$  and  $F_r$  of fragments in a fragmentation  $\mathcal{F}$ , their instances  $f_l$  and  $f_r$ , and the set  $\mathcal{C}'$  of confidentiality constraints completely covered by them, that is  $\forall c \in \mathcal{C}, c \in \mathcal{C}'$  if  $c \subseteq F_l \cup F_r$ . As an example, consider fragments  $F_2$  and  $F_3$  in the fragmentation in Figure 8.3 and the confidentiality constraints in Figure 8.2b,  $\mathcal{C}' = \{c_5, c_6\}$ . Since a loose association between  $F_l$  and  $F_r$  provides information about associations among groups of tuples in the fragments, the first step necessary to define a loose association consists in partitioning the tuples in  $f_l$  and  $f_r$  into groups. We note that, as described in the following, the size of the groups into

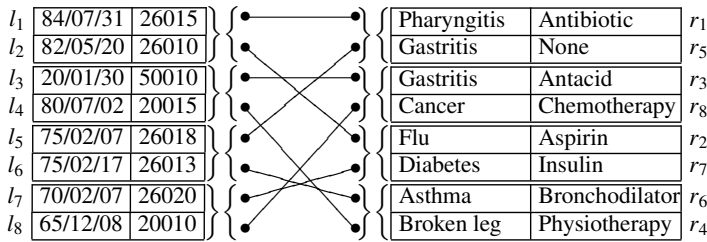


Fig. 8.7: An example of (2,2)-grouping.

which tuples in  $f_l$  and  $f_r$  are partitioned impacts the privacy degree guaranteed by the loose association. Therefore, in [DFJ<sup>+</sup>10a], the authors associate a parameter  $k$  with grouping, stating the lower bound to the size of groups. A  $k$ -grouping partitions the tuples in a fragment  $f$  in groups of size greater than or equal to  $k$ . A  $k$ -grouping is *minimal* if it minimises the size of each group (or, equivalently, maximises the number of groups), while respecting the lower threshold  $k$ . We note that the grouping parameter used for  $F_l$  can possibly be different from the grouping parameter used for  $F_r$ . Notation  $(k_l-k_r)$ -grouping is used to represent a  $k_l$ -grouping over  $F_l$  and a  $k_r$ -grouping over  $F_r$ . It is *minimal* if both the  $k_l$ -grouping over  $F_l$  and the  $k_r$ -grouping over  $F_r$  are minimal. On the basis of the  $(k_l-k_r)$ -grouping defined over  $F_l$  and  $F_r$ , it is possible to define a *group association*  $A$ , representing the relationships between the tuples in  $f_l$  and  $f_r$  at the group level. Intuitively, for each tuple  $t$  in the original relation  $r$ ,  $A$  includes a tuple representing the relationship between the group identifiers assigned by the  $(k_l-k_r)$ -grouping to the semi-tuple  $l$  representing  $t$  in  $f_l$  and to the semi-tuple  $r$  representing  $t$  in  $f_r$ . As an example, Figure 8.7 graphically illustrates the group association, also represented as a relational table in Figure 8.8, induced by a (2,2)-grouping on fragments  $F_l = F_2$  and  $F_r = F_3$  in Figure 8.3.

The protection offered by publishing group-level associations can be compromised if the tuples within a group have the same value for the attributes in a confidentiality constraint. To capture this situation, in [DFJ<sup>+</sup>10a] the authors introduce the definition of *aliqueness* between the tuples in a fragment, with respect the confidentiality constraints  $\mathcal{C}'$  completely covered by  $F_l$  and  $F_r$  (which can possibly be exposed by the release of a  $(k_l-k_r)$ -grouping). Two tuples  $l_i$  and  $l_j$  in  $f_l$  ( $r_i$ ,  $r_j$  in  $f_r$ , respectively) are *alike*, denoted  $l_i \simeq l_j$  ( $r_i \simeq r_j$ , respectively), if there exists at least a constraint  $c$  in  $\mathcal{C}'$  such that  $l_i[c \cap F_l] = l_j[c \cap F_l]$  ( $r_i[c \cap F_r] = r_j[c \cap F_r]$ , respectively). For instance,  $r_3 \simeq r_5$  in Figure 8.7, since  $r_3$  and  $r_5$  assume the same value for attribute `Illness`, which is the unique attribute in  $c_5 \in \mathcal{C}'$  appearing in  $F_r$ .

A group association  $A$  enjoys a degree  $k$  of protection if every tuple in  $A$  indistinguishably corresponds to at least  $k$  distinct associations among tuples in the fragments. Practically,  $A$  is *k-loose* if for each group  $g_l$  in the left fragment (group  $g_r$  in the right fragment, respectively), the union of the tuples in all the groups with which  $g_l$  ( $g_r$ , respectively) is associated in  $A$  is a set that has cardinality at least  $k$  and that does not contain any tuples that are alike.

$F_l$			$A$		$F_r$		
DoB	ZIP	G	$G_l$	$G_r$	Illness	Treatment	G
$l_1$	84/07/31	26015	dz1	it1	Pharyngitis	Antibiotic	it1 $r_1$
$l_2$	82/05/20	26010	dz1	it3	Flu	Aspirin	it3 $r_2$
$l_3$	20/01/30	50010	dz2	it2	Gastritis	Antacid	it2 $r_3$
$l_4$	80/07/02	20015	dz2	it4	Broken Leg	Physiotherapy	it4 $r_4$
$l_5$	75/02/07	26018	dz3	it1	Gastritis	None	it1 $r_5$
$l_6$	75/02/17	26013	dz3	it4	Asthma	Broncodilator	it4 $r_6$
$l_7$	70/05/04	26020	dz4	it3	Diabetes	Insulin	it3 $r_7$
$l_8$	65/02/08	20010	dz4	it2	Cancer	Chemotherapy	it2 $r_8$

(a)

(b)

(c)

Fig. 8.8: An example of 4-loose association.

We note that there is a correspondence between the parameters  $k_l$  and  $k_r$  of the groupings and the degree  $k$  of looseness guaranteed by the group association induced by a  $(k_l, k_r)$ -grouping. It is seen immediately that a  $(k_l, k_r)$ -grouping cannot induce a  $k$ -loose association for a  $k > k_l \cdot k_r$ . To guarantee that a  $(k_l, k_r)$ -grouping induces a  $k$ -loose association with  $k = k_l \cdot k_r$ , in [DFJ<sup>+</sup>10a] the authors define three *heterogeneity properties*, proving that if the  $(k_l, k_r)$ -grouping satisfies these properties, the induced group association is  $k$ -loose with  $k = k_l \cdot k_r$ . The heterogeneity properties can be summarised as follows:

- *group heterogeneity*: a group cannot contain two tuples that are alike with respect to the constraints in  $\mathcal{C}^l$ ;
- *association heterogeneity*: a group in the left (right, respectively) fragment cannot be associated more than once with the same group in the right (left, respectively) fragment;
- *deep heterogeneity*: a group in the left (right, respectively) fragment cannot be associated with two groups in the right (left, respectively) fragment that contain alike tuples.

We note that a  $k$ -loose association is also  $k'$ -loose for any  $k' \leq k$ . To maximise utility in data release, while satisfying the privacy requirement given by parameter  $k$ , it is convenient to determine the  $(k_l, k_r)$ -grouping that guarantees  $k$ -looseness while minimising the size of groups. A  $(k_l, k_r)$ -grouping induces a *minimal* group association if  $A$  is  $k$ -loose and there does not exist a  $(k'_l, k'_r)$ -grouping, inducing a  $k$ -loose association, such that  $k'_l \cdot k'_r < k_l \cdot k_r$ .

*Example 8.7.* Consider the (2,2)-grouping of Figure 8.7. It is minimal and satisfies group, association, and deep heterogeneity. As a consequence, it induces a minimal 4-loose association, illustrated in Figure 8.8.

Publishing loose associations provides some information on the possible combinations of tuples in the different fragments, as it restricts the possible combinations to those allowed by the loose associations. As shown in [DFJ<sup>+</sup>10a], this permits

an increase in the precision of query evaluation, without violating confidentiality constraints.

## 8.7 Conclusions

The recent trend toward the public release and the outsourcing of large data collections, possibly including sensitive information, has required the definition of novel protection techniques and of specific privacy metrics to assess their effectiveness. In this chapter, we illustrated new privacy metrics based on information theoretic concepts, able to measure the contribution to the risk of disclosure of each released record. We then described privacy protection techniques designed to protect arbitrary sensitive associations through fragmentation, possibly combined with encryption. We finally discussed how data utility can be improved by publishing a sanitised version of associations broken by fragmentation.

Privacy protection in data publishing is an interesting emerging scenario. Therefore, many issues still need to be investigated, such as: the definition of metrics that take into consideration the purpose of data release; the development of solutions for the automatic definition of confidentiality constraints; and the consideration of data dependencies.



## Chapter 9

# Selective Exchange of Confidential Data in the Outsourcing Scenario

Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati

**Abstract** The evolution of information and communication technologies (ICTs) has introduced new ways for sharing and disseminating user-generated content through remote storage, publishing, and disseminating services. From an enterprise oriented point of view, these services offer cost effective and reliable data storage features that any organisation can take advantage of without long setup delays and capital expenses. Also, from an end-user point of view, distributed and shared data storage services offer considerable advantages in terms of reliability and constant availability of data. While on one hand data sharing services encourage and enhance the collaboration among users, on the other hand they need to provide proper protection of data, possibly enforcing access restrictions defined by the data owner. In this chapter, we present an approach for allowing users to delegate to an external service the enforcement of the access control policy on their resources, while at the same time not requiring complete trust in the external service. Our solution relies on the translation of the access control policy into an equivalent encryption policy on resources, and on a hierarchical key structure that exploits the relationships between groups or users. In this way, we limit both the number of keys to be maintained and the amount of encryption to be performed, while keeping a good flexibility with respect to policy updates and revocations.

## 9.1 Introduction

Nowadays, users are resorting more and more to external services for disseminating and sharing resources they want to make available to others. The outsourcing of storage and computation to external parties then promises to become a crucial component of future ICT architectures. Many businesses have already invested in this direction, meeting significant success. This evolution is justified by clear technological and economic trends. The correct administration and configuration of computing systems is expensive and presents large economies of scale, support-



ing the centralisation of resources. This is particularly significant when considering reliability and availability requirements, which are difficult to satisfy for final users and small/medium organisations. This trend towards outsourcing is testified by the large success of different kinds of services offering remote storage and backup (e.g., Box.net and Mozy) and services allowing a large open community of users to store and exchange resources (e.g., YouTube and Flickr). These services assume that the service provider is completely trusted and always entitled to access the resources. In many scenarios, however, the service is considered *honest-but-curious*, meaning that it is relied upon for the availability of outsourced data but it is not authorised to access the actual data content. The solutions proposed in the literature to address this problem assume that the data owner encrypts her data before outsourcing them and communicates the encryption key only to authorised users [HIM02b]. While these solutions effectively provide confidentiality of the outsourced data, they assume that any authorised user can access all the outsourced resources. Today users are more and more demanding solutions for regulating the publication and disclosure of their own content. To the aim of enforcing selective access to encrypted resources, recently novel approaches integrating access control and encryption have been proposed [DFJ<sup>+</sup>10c, DFJ<sup>+</sup>10b]. These approaches are based on selective encryption, which nicely combines data encryption with access control enforcement (see [Figure 9.1](#)). Selective encryption basically consists in encrypting different pieces of information using different keys and in selectively distributing encryption keys to users. Each user (which should be properly authenticated [CGP<sup>+</sup>08, GLM<sup>+</sup>04, GPSS05]) can decrypt, and therefore access, all and only the data for which she knows the encryption key. We note that users in the system can act as both data owner and data consumer. As a consequence, to limit the number of keys in the system and enjoy a unique encryption policy, the approach in [DFJ<sup>+</sup>10b] defines a key agreement solution, based on Diffie-Hellman technique. Although selective encryption guarantees access control policy enforcement, it does not efficiently support policy updates. To overcome this problem, novel solutions allow delegating to the service provider the complete management, not only the enforcement, of the authorisation policy [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c].

In this chapter, we illustrate how access control policies can be translated into equivalent encryption policies, guided by the principles of releasing at most one key to each user, and encrypting each resource at most once. We also describe a novel approach for enforcing policy updates, with the goal of limiting the data owner intervention. The remainder of this chapter is organised as follows. Section 9.2 presents some basic concepts. Section 9.3 describes how to define an encryption policy that correctly enforces the access control policy defined by the data owner. Section 9.4 presents how resources are published and accessed. Section 9.5 compares the solution illustrated in this chapter with the a PGP key-management strategy, showing the efficiency of our proposal. Section 9.6 analyses the security issues of our model. Section 9.7 illustrates how authorisation policy changes can be outsourced, while leaving to the data owner the control on policy management. Finally, Section 9.8 concludes the chapter.

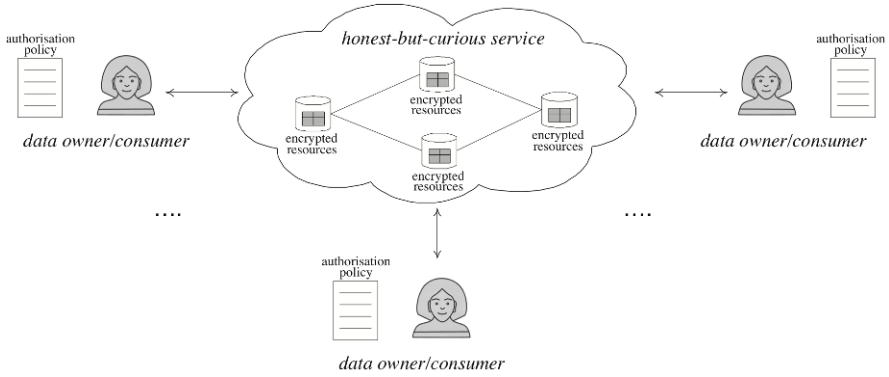


Fig. 9.1: Reference scenario.

## 9.2 Preliminaries

Let us assume a set of users  $\mathcal{U}$  who wish to selectively share their resources among themselves. Each user can act as either a *data owner* or a *data consumer*. A user plays the role of data owner when she makes her resources available to other users in the system. A user plays the role of data consumer when she requires access to resources owned by others. The sharing process is clearly selective since each resource might be accessible only to a subset of users in  $\mathcal{U}$ , as defined by the data owner. A shared resource can be modified only by its owner. Whenever a data owner wishes to share a resource with other users in the system, the management of the resource is delegated to an *external service* provider. The provider is trusted for the resource management, but it should not be allowed to either act on authorisation policies or access the resource content (*honest-but-curious* service). Also, the service provider is supposed not to prevent any authorised user from accessing her resources. In the following, given a user  $u \in \mathcal{U}$ ,  $R_u$  denotes the set of resources for which  $u$  is the owner, whilst  $\mathcal{R}$  denotes the set of resources managed by the service. Notation  $owner(r)$ , with  $r \in \mathcal{R}$ , represents the owner  $u \in \mathcal{U}$  of  $r$ . Each user  $u$  defines an *authorisation policy* to regulate access to her own resources.

**Definition 9.1 (Authorisation policy).** Given a data owner  $u \in \mathcal{U}$ , the *authorisation policy* defined by  $u$  over  $R_u$ , denoted  $P_u$ , is a set of pairs of the form  $\langle u_i, r_j \rangle$ , where  $u_i \in \mathcal{U}$  and  $r_j \in R_u$ .

The semantics of an authorisation  $\langle u_i, r_j \rangle \in P_u$  are that data owner  $u$  has granted to user  $u_i$  the permission to access resource  $r_j$ . Given a resource  $r \in \mathcal{R}$ , with  $u = owner(r)$ ,  $acl(r)$  denotes the *access control list* of  $r$ , that is, the set of users that can access  $r$  according to the authorisation policy  $P_u$ . The data owner  $u$  is always a member of the access control lists resulting from her authorisation policies.

*Example 9.1.* Consider a system where  $\mathcal{U} = \{A, B, C, D, E\}$ ,  $R_A = \{r_1, r_2\}$ ,  $R_B = \{r_3, r_4\}$ ,  $R_C = \{r_5\}$ ,  $R_D = R_E = \emptyset$ . The authorisation policies defined by  $A$ ,  $B$ , and  $C$  are:  $P_A = \{\langle A, r_1 \rangle$ ,

$\langle B, r_1 \rangle, \langle A, r_2 \rangle, \langle B, r_2 \rangle, \langle C, r_2 \rangle\}; P_B = \{\langle B, r_3 \rangle, \langle D, r_3 \rangle, \langle E, r_3 \rangle, \langle A, r_4 \rangle, \langle B, r_4 \rangle, \langle C, r_4 \rangle\};$   
 $P_C = \{\langle A, r_5 \rangle, \langle B, r_5 \rangle, \langle C, r_5 \rangle, \langle D, r_5 \rangle, \langle E, r_5 \rangle\}.$

According to these policies,  $acl(r_1) = \{A, B\}$ ,  $acl(r_2) = \{A, B, C\}$ ,  $acl(r_3) = \{B, D, E\}$ ,  
 $acl(r_4) = \{A, B, C\}$ , and  $acl(r_5) = \{A, B, C, D, E\}$

The goal is to realise a mechanism that allows data owners to securely share their resources only with authorised users, while preventing even the service provider from accessing a resource content. To this end, we employ an encryption scheme to enforce an authorisation policy. Encryption is applied with two objectives in mind: *i*) the efficiency, in order to minimise the number of keys managed by each user in the system; *ii*) the correct enforcement of the authorisation policy defined by the owner, in order to guarantee that a resource  $r$  is accessible uniquely to the users included in the corresponding  $acl(r)$ . The proposed encryption scheme enables any user to manage a *single secret* and any resource to be stored in one copy through the use of a single cryptographic key. A few public tables are stored on the external server and a user requesting access to a resource has to use a protocol that selectively returns the corresponding encryption key.

### 9.3 Encryption Schema

A peculiar characteristic of this scenario is that there are different owners responsible for different portions of the resources publicly available. A simple solution for addressing this aspect and for sharing resources in a selective way consists in applying the approaches developed for the data-outsourcing scenario [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c], where a single owner, before outsourcing her resources, encrypts them with different keys and each authorised user has a key from which she can derive all the keys of the resources she is authorised to access. Despite being simple, these solutions require each user to manage a large number of keys (potentially, one key for each data owner). We propose a novel solution that combines two cryptographic techniques: a *key agreement method* to share a secret key between a pair of users; a *key derivation method* that employs the secret keys shared between a pair of users to derive all keys used for encrypting resources that they are authorised to access. The integration of these two techniques results in an *encryption policy* that correctly enforces the authorisation policies  $P_u$ , for all  $u \in \mathcal{U}$  (see Section 9.3.3).

#### 9.3.1 Key Agreement

The key agreement method enforces a slight variation of the Diffie-Hellman (DH) key agreement method in such a way that two users taking part in a communication agree on a common secret through interacting with the external service. Our variation of the DH method works as follows. Let  $(\mathbb{G}, \cdot)$  be a public algebraic cyclic group of prime order  $q = |\mathbb{G}|$  and  $\cdot$  be the internal operation of the group with

multiplicative notation. We assume that  $\mathbb{G}$  is generated by an element  $g \in \mathbb{Z}_p$  (with  $p = 2q + 1$  and  $p, q$  two prime integers) in such a way that  $q = |\mathbb{G}|$  and  $\mathbb{G} = \{g^e \bmod p : 0 \leq e \leq q - 1\}$ . Each user  $u \in \mathcal{U}$  chooses a secret integer parameter  $e_u \in [0, q - 1]$ , computes the value  $g^{e_u} \in \mathbb{G}$ , and inserts  $g^{e_u}$  in a public catalogue managed by the external service that keeps track of the public parameters  $g$  and  $q$ . Whenever user  $u$  needs to share a common secret with user  $u_i$ , she can efficiently compute such a secret by querying the public catalog to retrieve the public parameters  $g^{e_{u_i}}$  and  $q$ , and by applying the following *key agreement function*.

**Definition 9.2 (Key agreement function).** Given a set  $\mathcal{U}$  of users, a set  $\mathcal{K}$  of keys, and a public algebraic cyclic group  $(\mathbb{G}, \cdot)$  of prime order  $q$ , with generator  $g \in \mathbb{G}$ , the *key agreement function* of a user  $u \in \mathcal{U}$  is a function  $ka_u : \mathbb{G} \mapsto \mathcal{K}$  that takes the public parameter  $g^{e_{u_i}} \in \mathbb{G}$  of a user  $u_i \in \mathcal{U}$  as input and returns the common secret between  $u$  and  $u_i$  computed as:  $ka_u(g^{e_{u_i}}) = (g^{e_{u_i}})^{e_u}$ .

Note that according to Definition 9.2, for all pairs of users  $u_i, u_j \in \mathcal{U}$ ,  $u_i \neq u_j$ ,  $ka_{u_i}(g^{e_{u_j}}) = ka_{u_j}(g^{e_{u_i}})$ . In the following, notation  $\mathcal{K}_{\mathcal{A}}$  is used to denote the set of key agreement functions of all users in  $\mathcal{U}$ . Intuitively, the evaluation of the key agreement function of a user  $u$  with respect to any other user  $u_i \in \mathcal{U}$  returns a secret key that can be exploited to encrypt the resources that should be accessible only to  $u$  and  $u_i$ . For instance, since according to authorisation policy  $P_A$  in Example 9.1, resource  $r_1$  is accessible only to  $A$  and  $B$ , user  $A$  can first encrypt  $r_1$  by using  $ka_A(g^{e_B})$  and can then deliver the encrypted resource to the external service. By generalising, a simple but at the same time inefficient solution for allowing a user  $u$  to share a resource  $r$  with  $n$  users  $u_1, \dots, u_n$ , consists in computing  $n$  keys  $ka_u(g^{e_{u_i}})$ , and in creating  $n$  copies of resource  $r$  encrypted with  $ka_u(g^{e_{u_i}})$ ,  $i = 1, \dots, n$ .

### 9.3.2 Key Derivation

A key derivation method allows the computation of a key starting from the value of another key and a publicly available piece of information called *token*. Given a set  $\mathcal{K}$  of keys and  $k_i, k_j \in \mathcal{K}$ , a token  $t_{i,j}$  between them is defined as  $t_{i,j} = E_{k_i}(k_j)$ , where  $E$  is a symmetric encryption function.<sup>1</sup> Each user knowing  $k_i$  can derive  $k_j$  by simply decrypting  $t_{i,j}$  with  $k_i$ . A chain of tokens is a sequence  $t_{i,1}, \dots, t_{n,j}$  such that  $t_{c,d}$  directly follows  $t_{a,b}$  in the chain only if  $b = c$ . The concept of key derivation via chains of tokens is formally captured by the following definition of *key derivation function*.

**Definition 9.3 (Key derivation function).** Given a set  $\mathcal{K}$  of keys, and a set  $\mathcal{T}$  of tokens, the *direct key derivation function*  $\tau : \mathcal{K} \mapsto 2^{\mathcal{K}}$  is defined as  $\tau(k_i) = \{k_j \in \mathcal{K} : \exists t_{i,j} \in \mathcal{T}\}$ . The *key derivation function*  $\tau^* : \mathcal{K} \mapsto 2^{\mathcal{K}}$  is a function such that  $\tau^*(k_i)$  is the set of keys derivable from  $k_i$  by chains of tokens, including the key itself (chain of length 0).

<sup>1</sup> Tokens can be defined according to different strategies (e.g., [AFB05]).

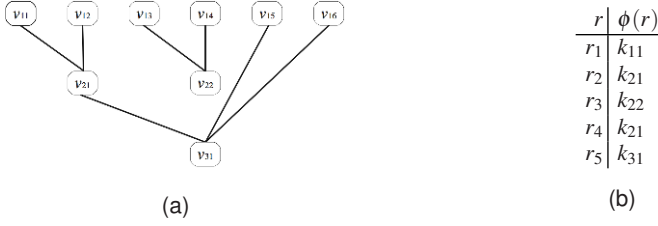


Fig. 9.2: An example of key and token graph (a) and key assignment function (b).

Graphically, a set  $\mathcal{K}$  of keys and a set  $\mathcal{T}$  of tokens can be represented via a *key and token graph*, with a vertex  $v_i$  for each key  $k_i \in \mathcal{K}$ , and an arc  $(v_i, v_j)$  for each token  $t_{i,j} \in \mathcal{T}$ . We call *root* a vertex in the key and token graph that does not have incoming arcs (i.e., a vertex whose key cannot be derived via tokens). Chains of tokens correspond to paths in the graph, and the key derivation function  $\tau^*(k)$  associates with each key  $k \in \mathcal{K}$  the keys of vertices reachable from the vertex associated with  $k$  in the graph. Figure 9.2(a) illustrates an example of key and token graph, where notation  $v_{ij}$  is used to denote the  $j$ -th vertex (from left to right) on the  $i$ -th level of the graph, and  $k_{ij}$  denotes the key associated with vertex  $v_{ij}$ . The root vertices of the graph are at level 1 (i.e.,  $v_{1j}$ ,  $j = 1, \dots, 6$ ). Note that for readability of the figure, arrows do not appear in the graph. The graph is oriented from top to bottom. The definition of tokens can support the general goal of encrypting resources without introducing redundancy in their management. The idea is that whenever a resource  $r$ , with  $u = \text{owner}(r)$ , must be accessible to  $n$  users  $u_1, \dots, u_n$ , the owner  $u$  can encrypt  $r$  with a key  $k \in \mathcal{K}$  and can compute a set of tokens that each user  $u_i$ ,  $i = 1, \dots, n$ , can then use for deriving key  $k$ . For instance, according to the authorisation policy  $P_A$  in Example 9.1, user  $A$  can encrypt her resource  $r_2$  with a key  $k \in \mathcal{K}$  and then can define two tokens, from  $ka_A(g^{e_B})$  to  $k$  and from  $ka_A(g^{e_C})$  to  $k$ , that users  $B$  and  $C$ , respectively, can exploit for deriving  $k$ . A *key assignment function* computes the keys used for encrypting resources.

**Definition 9.4 (Key assignment function).** Given a set  $\mathcal{R}$  of resources and a set  $\mathcal{K}$  of keys, the *key assignment function*  $\phi : \mathcal{R} \mapsto \mathcal{K}$  associates with each resource  $r \in \mathcal{R}$  the (single) key with which the resource is encrypted.

Figure 9.2(b) illustrates an example of key assignment function defined over the resources of Example 9.1. It is easy to see that the key used for encrypting  $r_5$  (i.e.,  $k_{31}$ ) can be derived from keys  $k_{11}$ ,  $k_{12}$ ,  $k_{15}$ , and  $k_{16}$ .

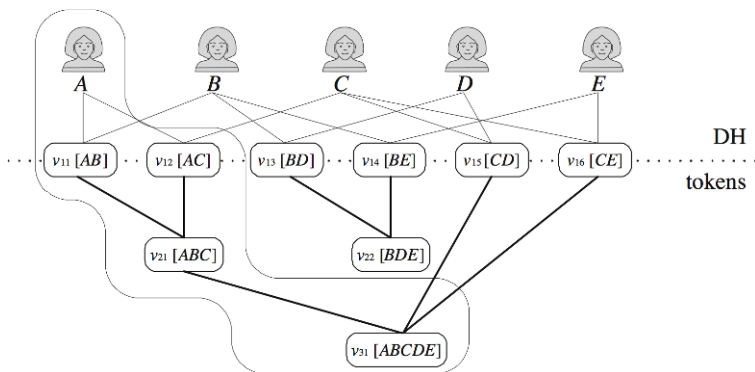


Fig. 9.3: An encryption policy graph.

### 9.3.3 Encryption Policy

An encryption policy regulates which resources are encrypted with which keys and which keys can be directly or indirectly computed by which users. Formally, an encryption policy is defined as follows.

**Definition 9.5 (Encryption policy).** Given a set  $\mathcal{U}$  of users and a set  $\mathcal{R}$  of resources, an *encryption policy* over  $\mathcal{U}$  and  $\mathcal{R}$ , denoted  $\mathcal{E}$ , is a 6-tuple of the form  $\langle \mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{T}, \mathcal{K}_{\mathcal{A}}, \phi \rangle$ , where  $\mathcal{K}$  is a set of keys,  $\mathcal{T}$  is a set of tokens defined over  $\mathcal{K}$ ,  $\mathcal{K}_{\mathcal{A}}$  is the set of key agreement functions of all users in  $\mathcal{U}$ , and  $\phi$  is a key assignment function.

An encryption policy can be represented via a graph, called *encryption policy graph*, obtained from the key and token graph corresponding to  $\mathcal{K}$  and  $\mathcal{T}$  by adding a vertex for each user  $u \in \mathcal{U}$ , and by adding an edge from each vertex representing  $u$  to vertices representing keys  $ka_u(g^{e_{u_i}})$ , for all  $u_i \in \mathcal{U}$ ,  $u \neq u_i$ . The vertices representing pairs of users are inserted in the graph if and only if there is at least one token starting from them. Figure 9.3 illustrates an example of encryption policy graph, where each vertex has been labelled with the set of users who know or can derive the corresponding key, thick edges represent tokens, and thin edges represent the computations of the key agreement functions. Note that the information about the users who can derive the key associated with a specific vertex does not necessarily coincide with the real identities of the users. As a matter of fact, each user can be identified via a pseudonym that may be selected by the user herself. Also, the root vertices of the key and token graph are the vertices representing the keys computed through the key agreement functions in  $\mathcal{K}_{\mathcal{A}}$ ; these keys need to be directly computed by the users and do not exploit tokens. It is easy to see that each user  $u$  can directly or indirectly compute the keys associated with vertices along the paths starting from the vertex representing  $u$ . The first step is always a Diffie-Hellman computation whose resulting key is the starting point of the token chains followed by the user. Formally, the set of keys that a user can derive is defined as follows.

USER		RESOURCE				TOKEN		
user_id	public	res_id	owner	label	enc_res	source	destination	token_value
A	$g^{e_A}$	$r_1$	A	AB	$\alpha$	AB	ABC	$E_{k_{11}}(k_{21})$
B	$g^{e_B}$	$r_2$	A	ABC	$\beta$	AC	ABC	$E_{k_{12}}(k_{21})$
C	$g^{e_C}$	$r_3$	B	BDE	$\delta$	BD	BDE	$E_{k_{13}}(k_{22})$
D	$g^{e_D}$	$r_4$	B	ABC	$\varepsilon$	BE	BDE	$E_{k_{14}}(k_{22})$
E	$g^{e_E}$	$r_5$	C	ABCDE	$\zeta$	CD	ABCDE	$E_{k_{15}}(k_{31})$
						CE	ABCDE	$E_{k_{16}}(k_{31})$
						ABC	ABCDE	$E_{k_{21}}(k_{31})$

Fig. 9.4: An encryption policy catalogue.

**Definition 9.6 (User keys).** Given an encryption policy  $\mathcal{E} = \langle \mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{T}, \mathcal{K}_A, \phi \rangle$ , the set of keys that a user  $u \in \mathcal{U}$  can compute, denoted  $K_u$ , is defined as

$$K_u = \bigcup \tau^*(ka_u(g^{e_{u_i}})) : u_i \in \mathcal{U}, u_i \neq u, \text{ and } ka_u(g^{e_{u_i}}) \in \mathcal{K}.$$

Each user  $u$  can then access any resource  $r$  such that  $\phi(r) \in K_u$ . For instance, with respect to the encryption policy graph in Figure 9.3, the portion of the graph that user A can exploit for key derivation is delimited by a continuous line and contains the set  $K_A = \{k_{11}, k_{12}, k_{21}, k_{31}\}$  of keys she can compute. Our goal is then to translate the authorisation policies defined by the users in  $\mathcal{U}$  into a *correct* encryption policy  $\mathcal{E}$ . The concept of encryption policy correctness is formally defined as follows.

**Definition 9.7 (Correctness).** Given a set  $\mathcal{U}$  of users, a set  $\mathcal{R}$  of resources, a set  $\mathcal{P} = \bigcup_{u \in \mathcal{U}} P_u$  of authorisation policies, and an encryption policy  $\mathcal{E} = \langle \mathcal{U}, \mathcal{R}, \mathcal{K}, \mathcal{T}, \mathcal{K}_A, \phi \rangle$  over  $\mathcal{U}$  and  $\mathcal{R}$ , we say that  $\mathcal{E}$  *correctly enforces*  $\mathcal{P}$  iff the following conditions hold:

$$\forall u \in \mathcal{U}, r \in \mathcal{R} : \phi(r) \in K_u \Rightarrow \langle u, r \rangle \in \mathcal{P} \quad (\text{Soundness})$$

$$\forall u \in \mathcal{U}, r \in \mathcal{R} : \langle u, r \rangle \in \mathcal{P} \Rightarrow \phi(r) \in K_u \quad (\text{Completeness})$$

The encryption policy graph in Figure 9.3 and the key assignment function in Figure 9.2(b) represent a correct encryption policy given the authorisation policies in Example 9.1. To allow users to derive the keys needed for accessing the resources, a portion of the encryption policy must be publicly available from the external service responsible for resource management. This public information is represented as a *catalogue* composed of three tables: USER, RESOURCE, and TOKEN.

Table USER contains a tuple for each user in  $\mathcal{U}$  and has two attributes: *user\_id* is the user identifier and *public* is the public parameter (Diffie-Hellman public key) of the user. Table RESOURCE includes a tuple for each resource in  $\mathcal{R}$  and has four attributes: *res\_id* is the resource identifier; *owner* is the identifier of the user who published the resource; *label* is the label of the vertex in the encryption policy graph whose corresponding key is  $\phi(r)$ ; and *enc\_res* is the encrypted copy of the resource, whose content has been preemptively signed by the data owner.

Table TOKEN contains a tuple for each token in  $\mathcal{T}$  and is characterised by three attributes: *source* and *destination* are the labels of the corresponding source and



destination vertices in the encryption policy graph; *token\_value* is the token value computed as  $E_{k_{source}}(k_{destination})$ .

Figure 9.4 illustrates the public catalogue corresponding to the key assignment function in Figure 9.2(b) and the encryption policy graph in Figure 9.3. In order to allow users to verify the authenticity of resources, without relying on the trusted behaviour of the server, we assume resources to be signed by their owner. To this end, we adopt the DSA signature scheme: 1) Diffie-Hellman  $e_u$  and  $g^{e_u}$  parameters can be used as DSA private and public key, respectively; and 2) the Diffie-Hellman public parameters  $g$  and  $q$  (Section 9.3.1) can be chosen to satisfy the security criteria needed to use them also as DSA public parameters. Before outsourcing a resource  $r$ ,  $owner(r)$  computes the plaintext digest of  $r$  by applying a cryptographic hash function  $h$  to the plaintext content of  $r$ . She then signs  $h(r)$  using her secret Diffie-Hellman parameter  $e_u$ , and encrypts both  $r$  and its signature. A user accessing  $r$  can check the signature of the resource, after its decryption, by using the public Diffie-Hellman parameter  $g^{e_u}$  of  $owner(r)$  and the publicly available cryptographic hash function  $h$ .

## 9.4 Resource Sharing Management

The defined approach provides the users with the functionality for *publishing* and *accessing* resources. The publish functionality allows data owners to compute the digest, sign, and correctly encrypt their resources prior to delivering them to the service provider. Each user  $u$  can operate on a subset of the whole encryption policy graph. As a matter of fact,  $u$  is able to first create and then use token chains whose starting points are the root vertices corresponding to keys that  $u$  can compute through Diffie-Hellman computations (i.e., root vertices representing  $u$  and another user in the system). Therefore, whenever user  $u$  needs to share a resource  $r$  with other users in the system, she must first encrypt  $r$  with a new key and then must add the appropriate tokens that the other users in  $acl(r)$  can use to derive the new key. The creation of these new tokens can exploit token chains previously created by  $u$  to reach vertices representing a subset of the users in  $acl(r)$ . Note that if there already exists a key only derivable by users in  $acl(r)$ ,  $u$  simply computes such a key through the appropriate token chain and then encrypts  $r$  with the derived key.

*Example 9.2.* Figure 9.5 illustrates the evolution of an encryption policy graph following a sequence of publish operations. There are five users,  $\mathcal{U} = \{A, B, C, D, E\}$ , and at the initial state, the authorisation policies already enforced through the hierarchy are:  $P_A = \{\langle A, r_1 \rangle, \langle B, r_1 \rangle, \langle A, r_2 \rangle, \langle B, r_2 \rangle, \langle C, r_2 \rangle\}$ ,  $P_B = \{\langle B, r_3 \rangle, \langle D, r_3 \rangle, \langle E, r_3 \rangle\}$ . Upon each publication, if there is no key for the involved  $acl$ , a new key is generated together with the tokens allowing derivation of the key from all the users in the  $acl$ . The figure also reports the keys that must be computed between pairs of users for ensuring such a derivation.

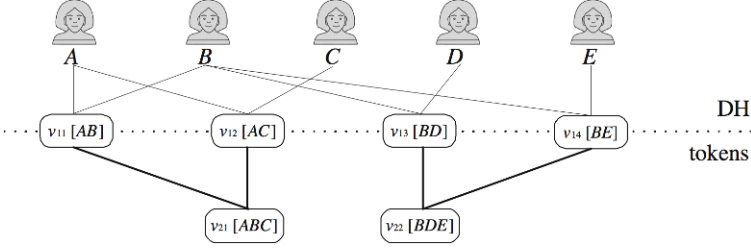


**Initial State:**  $P_A = \{\langle A, r_1 \rangle, \langle B, r_1 \rangle, \langle A, r_2 \rangle, \langle B, r_2 \rangle, \langle C, r_2 \rangle\}$ ,  
 $P_B = \{\langle B, r_3 \rangle, \langle D, r_3 \rangle, \langle E, r_3 \rangle\}$

---

**Request from B:**  $\text{Publish}(r_4, B, e_B, \{A, B, C\})$

---




---

**Request from C:**  $\text{Publish}(r_5, C, e_C, \{A, B, C, D, E\})$

---

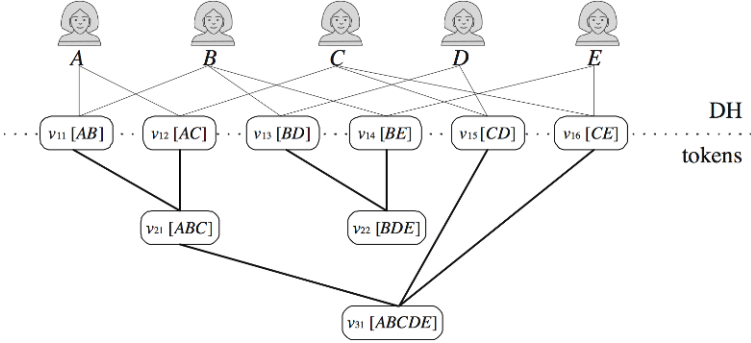


Fig. 9.5: A sequence of publishing operations.

The access functionality allows users to retrieve the resources that they are authorised to access and to verify their signature. In particular, every time an authorised user  $u$  needs to access a resource  $r$ , the service has to deliver the encrypted resource to  $u$  along with a token chain ending at the vertex representing  $acl(r)$ , which the user follows to derive the decryption key. User  $u$  can then decrypt the resource and use the public Diffie-Hellman parameter of  $owner(r)$  to verify the signature of  $r$ .

*Example 9.3.* Consider the state resulting from the sequence of operations of Example 9.2 and the request of user  $B$  to access  $r_2$ . Since  $B \in acl(r_2)$ , the access function extracts  $r_2^k$  from table RESOURCE, finds the shortest path from  $v_{11}$  to  $v_{21}$  (i.e., the vertex whose key is used for encrypting  $r_2$ ), and derives  $k_{21}$  from  $k_{11}$ . The resource  $r_2^k$  is decrypted using  $k_{21}$  and the resulting *signature* is verified through the public parameter of the resource's owner  $A$ .

## 9.5 Comparison with the PGP's Key-Management Strategy

An alternative solution for enforcing an authorisation policy via encryption could be based on a PGP-based approach.<sup>2</sup> Following this model, each user in the system has a public/private key pair and each resource  $r$  is encrypted with an arbitrary symmetric key  $k$ . For each user  $u$  in  $acl(r)$ ,  $owner(r)$  encrypts  $k$  with the public key of  $u$ . Both the encrypted resource and the encrypted copies of its encryption key are stored at the storage server. To access resource  $r$ , a user  $u \in acl(r)$  needs first to decrypt the symmetric encryption key  $k$  with her private key and then to decrypt the resource with  $k$ . Despite being simple, this approach does not scale well since a resource shared among a large community has to be extended with a large descriptor, containing the resource key encrypted with the public key of every user authorised to access the resource; while our solution exhibits lower storage usage. Suppose the encoding of any  $acl$  requires 4 bytes, the size of a symmetric key is 128-bit, and a symmetric key is encrypted with a 1024-bit public key thus obtaining an encryption block of 128 bytes. A PGP-based solution requires the addition of a new column in table RESOURCE for storing the resource descriptors. The additional storage space necessary to maintain these resource descriptors is  $128 \sum_{r \in \mathcal{R}} |acl(r)|$  bytes. Our approach instead requires the storage of table TOKEN, which is not needed with PGP. Since the size of each row of table TOKEN is 24 bytes (4 bytes for attribute *source*, 4 bytes for attribute *destination*, and 16 bytes for attribute *token\_value*, which corresponds to the ciphertext obtained by encrypting a symmetric key with a symmetric algorithm), the total storage space required for table TOKEN is  $24|\mathcal{T}|$  bytes, where  $|\mathcal{T}|$  corresponds to the number of edges in the key and token graph. In the worst case, each resource in  $\mathcal{R}$  has a different access control list composed of more than two users thus implying at most  $|\mathcal{R}|$  non-root vertices in the graph. Since each non-root vertex  $v$  representing a set  $acl$  of users, has at most  $|acl| - 1$  incoming edges, the storage space required by table TOKEN is less than  $24 \sum_{r \in \mathcal{R}} |acl(r)|$  bytes, which is 5.3 times less than the space required by a PGP-based solution. Consider now a system where  $acl(r_i) = \{u_1, \dots, u_{i+1}\}$  and  $owner(r_i) = u_1$ ,  $i = 1, \dots, |\mathcal{R}|$ . This configuration is the best case in terms of the storage space required for our approach: the space for storing the resource descriptors is  $128(|\mathcal{R}| + 1)(|\mathcal{R}| + 2)/2$  bytes and the storage space required for table TOKEN is  $24|\mathcal{T}| = 24(2|\mathcal{R}|)$  bytes since each vertex in the key and token graph has 2 incoming edges, which is clearly less than the space needed with a PGP-based solution. Another drawback of a PGP-based solution is that requiring each resource to be associated with its own descriptor does not allow for exploiting the fact that different resources might have the same access control list. The management of key updates is also difficult, since a public key update would require recomputing all of the descriptors of the corresponding resources. In our solution, this operation would instead require the update of tokens corresponding to the arcs outgoing from a few root vertices of the key and token graph.

<sup>2</sup> IETF–OpenPGP Working Group, RFC 4880, <http://www.openpgp.org/>

## 9.6 Exposure Evaluation

Threats in key-agreement or key-distribution schemes are realised by an adversary who may eavesdrop, replay, or substitute messages that are transmitted over a communication channel. In the considered scenario, each user accesses the storage service from different and independent locations, and the security threats are due to a subject (service or user) that aims at accessing resources with *acls* that do not include her. The encryption is assumed to be robust, thus improper accesses to a resource by unauthorised parties can only happen if a party improperly acquires the key with which the resource is encrypted. Impersonation of the service is a threat excluded from the analysis since the communication between the user and the service is assumed to be over an *SSL channel*. Also, the *key derivation* method is assumed to be *secure* since a user knowing a key  $k_i$  associated with a vertex  $v_i$  cannot compute keys associated with vertices that are not descendants of  $v_i$  in the encryption policy graph [AFB05, ADFM06]. The way a malicious party can get an access key is by *masquerading* as a legitimate user so that other users in the system are provided with the public DH parameter of the malicious party instead of the correct one. The problem of users claiming an identity they do not own (e.g. phony celebrity pages in Facebook and MySpace) lies outside of the technical realm. We are instead interested in the technical problem of preventing the service from behaving maliciously, thereby compromising the confidentiality of resources by presenting DH parameters that the service controls. The traditional techniques to overcome this threat consist in exchanging the public DH parameters adopting 1) an off-band user-to-user communication on a trusted channel, or 2) a solution involving one or more certification authorities. These techniques are well known and robust, but they are also known to be costly when each user of the system has to rely on them for their security. Also, an identity-based approach [BF03], despite being certificate-less, does not fit the requirements for our scenario because it basically shifts the user's trust from the storing service provider to a centralised Key Generation Center.

### 9.6.1 Anonymous Accesses

A novel and inexpensive possibility is based on the ability of users to query the public table *USER anonymously*. The service provider can easily monitor the communication channels employed by users and modify the content of the public tables. While the service provider appears extremely powerful, we assume the provider has a strong incentive to have a good reputation among users, since it is sufficient for a limited number of users to report improper behaviour of the service to have all other users lose confidence in it. According to this observation, our approach is based on the execution of random checks, by the users, on the honest behaviour of the service. Suppose a service provider  $S$  behaves maliciously with the goal of accessing a resource that a legitimate victim  $B$  is entitled to access. The attack can be directed to resources owned by a particular user  $A$  or to all resources that  $B$  could access. Let

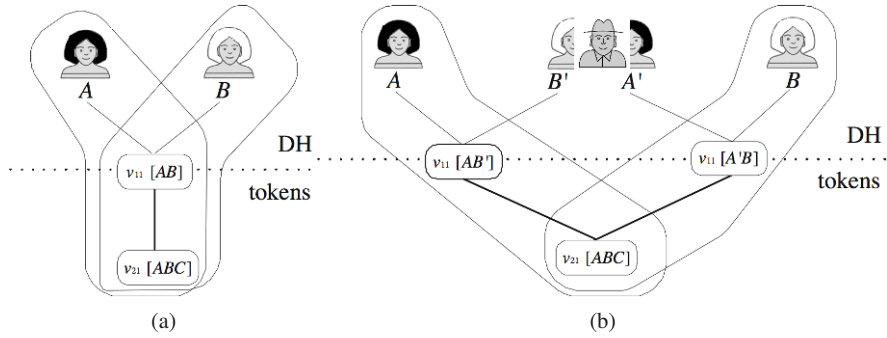


Fig. 9.6: Man-in-the-middle attack.

us consider first the attack aimed at acquiring the resources of a specific user  $A$ , and then generalise the treatment. To act as a *man-in-the-middle* (MitM), masquerading as  $B$  with respect to  $A$ , upon  $A$ 's request to retrieve the public parameter of  $B$  (i.e.,  $g^{e_B}$ ), the service  $S$  will have to respond with a fake public parameter  $g^{e'_B}$ . Then,  $S$  will be able to derive the common key between  $B$  and  $A$ , and therefore access resources whose *acl* is equal to, or contains  $\{A, B\}$ . However, to avoid being detected by  $B$ , the service should ensure  $B$ 's ability to access the resources that  $A$  wishes to share with  $B$ . This forces  $S$  to encrypt the resources also with a key that both  $S$  and  $B$  can compute. Hence,  $S$  will have to masquerade as  $A$  for  $B$  and will have to: 1) sign a copy of  $A$ 's resource (which  $S$  can now acquire) with a fake Diffie-Hellman private parameter  $e'_A$  (instead of the genuine  $e_A$ ); 2) encrypt the copy of  $A$ 's resource and its signature to be made accessible to  $B$ ; 3) provide to  $B$  the fake public parameter  $g^{e'_A}$  allowing  $B$  to verify the signature and determine the key with which the copies of the resources have been encrypted; 4) create a new copy of all the tokens that were supposed to originate from the authentic common key between  $B$  and  $A$  so that they originate from the fake key (actually agreed between  $B$  and the service). These tokens are needed to ensure  $B$  will be able to access all resources whose *acl* includes  $\{A, B\}$ . The MitM attack implies that the service will always have to return the fake  $B$ 's public parameter  $g^{e'_B}$  to  $A$ , and the fake  $A$ 's public parameter  $g^{e'_A}$  to  $B$ , while instead returning the correct  $g^{e_B}$  and  $g^{e_A}$  to other users. Note that due to the symmetric nature of the attack, by aiming at acquiring access to  $A$ 's resources accessible to  $B$ , the service also acquires access to  $B$ 's resources accessible to  $A$ . Note also that this symmetric behaviour makes the attack applicable only to pairs of users that have never shared resources before. If the service aims at accessing all the resources to which  $B$  has access,  $S$  should mount a similar attack for all the other users in the system. Figure 9.6 illustrates an example of an encryption policy graph in the case of a malicious service mounting a MitM attack between users  $A$  and  $B$ . Our protection relies on two easily enforceable assumptions. First, we assume that public parameter requests to the service can be made anonymously (e.g., via a proxy or a mixing protocol [DMS04]), so that the service will not be able to infer which

user is submitting the request. Second, we assume users can randomly query the service for their own Diffie-Hellman public key or for other keys they already know. With respect to our example, the service will not know if the request for  $B$ 's public parameter comes from a user different from  $A$ , including  $B$  (and therefore it should respond  $g^{e_B}$ ), or from  $A$  (and therefore it should respond  $g^{e'_B}$ ). While the service can try to guess the source of the request, it is reasonable to expect a non-negligible probability  $p_w$  of wrong guess. It is then possible to put constraints on the number of attacks that the service would be able to realise without being detected; with a simple statistical model, we obtain that  $\lceil 1/p_w \rceil$  checks will be sufficient to detect with at least  $1 - 1/e$  probability (0.632) the illicit behaviour.<sup>3</sup> The probability of the attack being detected quickly increases with the increase of the number of anonymous retrievals of keys. Hence, since the service has a strong incentive to keep its reputation intact, it is clearly driven to avoid the MitM attack. For systems aiming to serve a large community of users, we expect this protection to be able to offer a high degree of robustness.

## 9.7 Encryption Policy Updates

The encryption policy described in the previous sections assumes that keys and tokens are computed on the basis of existing authorisation policies before sending the encrypted resources to the server. While the authorisations set at initialisation time might not be changed too frequently, many situations require dynamic alterations to it in order to grant or revoke privileges to either new or old users, respectively. Therefore, every time an authorisation on a resource  $r$  is granted or revoked,  $acl(r)$  changes accordingly. In terms of encryption policy, this mandates a change of the key used to encrypt the resource, so that it will be accessible only to users in the modified  $acl$ . This operation requires decrypting the resource (with the key with which it is currently encrypted), to retrieve the original plaintext (since the owner may not keep a local copy of her outsourced data), and then re-encrypt it with the new key. Such an overhead, in terms of both communication and computation, for managing authorisation changes does not fit current scalability needs. Therefore, we defined mechanisms to outsource the also evolution of the authorisation policies. Note that this delegation is possible since the server is considered trustworthy, i.e., it is assumed to properly carry out the service, albeit it is not trusted to respect data confidentiality. The solution, called *Over-encryption* [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c], enforces policy changes on the encrypted resources themselves without the need of decrypting them, and may thus be performed by the server. The security goal consists in minimising the server gain in colluding with some user to get access to data owned by others.

<sup>3</sup> If  $p_w$  is equal to  $1/n$ , the probability for the server of guessing  $n$  times is  $(1 - 1/n)^n$ , which is a known mathematical series approximating  $e^{-1}$ .

### 9.7.1 Two-Layered Encryption Model

The encryption policy model is enriched with two layers of encryption:

A *Base Encryption Layer* (BEL) is applied by the data owner before transmitting data to the server in order to enforce the access policy existing at initialisation time, as described in the previous sections.

A *Surface Encryption Layer* (SEL) is applied by the service provider, over the resources already encrypted by the data owner, to enforce the dynamic changes over the policy.

Each user (as both data owner and data consumer) remains directly responsible only for her own Diffie-Hellman secret key, while the enforcement of the encryption policy requires her to perform, at SEL level, a key agreement with the service provider, and at BEL level, a non-interactive key agreement with the target data owner. In order to access the target resource, each user has to encrypt (respectively decrypt) the resource twice: firstly with the unique access key at SEL layer, and secondly with the key corresponding to the unique access key at BEL layer.

Two basic approaches can be followed in the construction of the two levels, called Full-SEL and Delta-SEL, having different performances and protection guarantees [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c].

*Full-SEL.* With this mode of encryption, the SEL policy is set to mimic the BEL policy: for each derivation key in BEL, a corresponding key is defined in SEL, and for each token in BEL, a corresponding token is defined in SEL.

*Delta-SEL.* With this mode of encryption, when a resource is uploaded for the first time, the SEL level does not add any additional protection, while enforcing a double encryption only when a change in the existing authorisation policies of a data owner is requested.

An approach where the authorisation enforcement is completely delegated at the SEL level, whilst the BEL one simply applies a uniform over-encryption to protect the plaintext content from the server's eyes (i.e., each data owner selects only one key, and distributes it to all the data consumers she wants to share her resource with), would present a significant exposure to collusion attacks. The Full-SEL always requires double encryption to be enforced (even when authorisations remain unchanged), thus doubling the decryption load of users for each access. However, in terms of efficiency, the use of a double layer of encryption, adds only a negligible computational overhead [DFJ<sup>+</sup>10c]. By contrast, the Delta-SEL approach requires double encryption only when actually needed to enforce a change in the authorisations. The reasons for choosing one of the two modes of operation over the other, are related to both the collusion threats and the data owner inferences on the granularity of the access control policies imposed on the resources she wants to share. Moreover, the assumptions on the evolution over time of these policies play a key role in the decision (see Section 9.7.3).

### 9.7.2 Over-Encryption

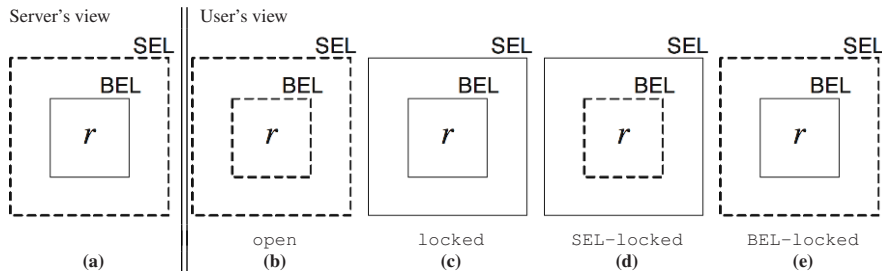
For each data owner  $O$ , the grant and revoke policy alterations are enforced through a two-layer encryption system, embodied in the **over-encrypt**( $R_O, acl(R_O)$ ) primitive, where  $R_O$  is the set of resources that must be accessible only to users in the corresponding access control list,  $acl(R_O)$ . The over-encrypt primitive manages the proper resource confinement through the SEL layer in order to deny the access to a resource to a certain subset of users, while it employs the BEL layer in order to establish which users are allowed to access it. This results into two independent access control actions, which are applied synergically in order to compose the desired access restrictions. For instance, if two resources  $r, r'$  share the same access control list (and thus the same BEL key), granting the access to  $r$  to a new user (i.e., allowing her to access  $r$  through the related BEL key) will result into gifting her with the right to access  $r'$ . In order to compensate this undesired privilege, the SEL encryption layer is updated to deny the access to  $r'$  to the new user, through over-encrypting  $r'$  with a new key, which can be derived only by the legitimate members of  $acl(r')$ . Consequently, revoking the access rights to a resource  $r''$  for a specific user is as simple as removing the user from  $acl(r'')$  and subsequently employing the SEL encryption layer in order to prevent her from accessing  $r''$ .

Depending on whether the Full- or Delta-SEL mode of encryption is employed, the over-encryption operation is performed in two different ways. In the Full-SEL case, the SEL encryption layer is employed to deny the access to a resource  $r$  to all the users but the ones in  $acl(r)$ . This is done regardless of the users being allowed or not to access  $r$ , through being able to derive the related BEL key. By contrast, in the Delta-SEL approach, the locking effect of the SEL layer is employed only in order to limit the access rights granted by the BEL keys owned by the no longer authorised users [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c].

### 9.7.3 Collusion Evaluation

We assume both the key derivation functions and the encryption primitives are semantically and provably secure [AFB05], even when combining the information available to many users. Moreover, we assume that each user correctly manages her keys. It still remains to evaluate whether the approach is vulnerable to attacks from users who access and store all information offered by the server, or from *collusion* attacks, where different users (or a user and the server) combine their knowledge to access resources they would not otherwise be able to access. Note that for collusion to exist, both parties should benefit from it, otherwise they will not have any incentive in colluding. In order to model the information leakage, we assume that users are not oblivious (i.e., they have the ability to store and keep indefinitely all information they were entitled to access). In order to examine the different views that each user can have on a resource  $r$ , we employ a graphical notation with resource  $r$  in the center and with fences around  $r$  denoting the barriers to the access



Fig. 9.7: Possible views on resource  $r$ .

imposed by the knowledge of the keys used for  $r$ 's encryption at both the BEL level (inner fence) and the SEL level (outer fence). The fence is continuous if there is no knowledge of the corresponding key and it is discontinuous otherwise. Figure 9.7(a) shows the view of the SEL server itself, which knows the SEL-level key, but does not have access to the BEL-level key. On the right, the *open* view corresponds to the view of authorised users, while the remaining ones (*locked*, *SEL-locked*, *BEL-locked*) show the views of non-authorised users.

Collusion can take place every time two entities, combining their knowledge (i.e., the keys known to them) can acquire knowledge that *neither* of them has access to. Therefore, users having the *open* view need not be considered as they have nothing to gain in colluding (they already access  $r$ ). Following the same line of reasoning, users having the *locked* view will not be considered, since they have nothing to offer. In the Full-SEL approach, no one but the server can have a *BEL-locked* view, while only a user can have an *SEL-locked* view. This describes the only possible threat of collusion, because the knowledge of the server allows for lowering the outer fence, while the knowledge of the user allows for lowering the inner fence.

There are only two reasons for which a user can have the *SEL-locked* view on a resource. 1) The user was previously authorised to access the resource and the authorisation was then revoked. In this case, since the user is supposed to be non-oblivious, she has no gain in colluding with the server. It is therefore legitimate to consider this case ineffective with respect to collusion risks.<sup>4</sup> 2) The user has been granted the authorisation for resource  $r'$  that was, at initialisation time, encrypted with the same key as  $r$  (i.e.,  $\text{acl}(r') \subseteq \text{acl}(r)$ ), leaving  $r$  *SEL-locked* [DFJ<sup>+</sup>07, DFJ<sup>+</sup>10c]. In this situation ( $r$  from *locked* to *SEL-locked*), the user has never had access to  $r$  and must not be able to gain it, therefore there is indeed exposure to collusion.

The Full-SEL approach provides superior protection, as it reduces the risk of exposure, which is limited to collusion with the server. By contrast, the Delta-SEL approach exposes also to single (planning-ahead) users. Therefore, each data owner, when choosing between the use of Delta-SEL or Full-SEL, should prefer the first

<sup>4</sup> We assume, without loss of generality, that any time a resource is updated, the data owner encrypts it with another BEL key as it were a new one



one when it is likely that: her access policy will be relatively static, sets of resources sharing the same *acl* at initialisation time represent a strong semantic relationship rarely split by policy evolution, or resources are grouped in the BEL in fine granularity components where most of the BEL nodes are associated with a single or few resources. Indeed, in these situations, the risk of information leakage due to collusion is limited also in the Delta-SEL approach. By contrast, if authorisations have a more dynamic and chaotic behaviour, the Full-SEL approach may be preferred to limit exposure due to collusion (necessarily involving the server). Also, the collusion risk may be minimised through a proper organisation of the resources to reduce the possibility of policy splits. This could be done either by employing a finer encryption granularity and/or better identifying resource groups characterised by a persistent semantic affinity (in both cases, using in the BEL different keys for resources with identical *acl*).

## 9.8 Conclusions

The outsourcing to honest-but-curious service providers of large data collections requires the definition of novel access control systems, to support the selective release of outsourced information to authorised users. In this chapter, we illustrated a novel approach combining authorisations and encryption to enforce access control policies defined by different data owners, who want to selectively share their resources. Access control policies translate into equivalent encryption policies, such that each user can decrypt all and only the resources she is authorised to access. We also described a solution that allows data owners to outsource to the service provider possible updates to the access control policies, by nicely combining two layers of encryption. We note that the access control system introduced in this chapter can be easily integrated with the proposal in [DFJ<sup>+</sup>08], to protect the confidentiality of the policy if it needs to be kept secret.

## References Part II

- [AA01] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS 2001)*, Santa Barbara, CA, USA, 2001.
- [ABG<sup>+</sup>05] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: a distributed architecture for secure database services. In *Proc. of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, Asilomar, CA, USA, January 2005.
- [ACBM08] Elli Androulaki, Seung Geol Choi, Steven M. Bellovin, and Tal Malkin. Reputation systems for anonymous networks. In *PETS '08: Proceedings of the 8th international symposium on Privacy Enhancing Technologies*, pages 202–218, Berlin, Heidelberg, 2008. Springer-Verlag.
- [Acc08] Rafael Accorsi. *Automated counterexample-driven audits of authentic system records*. PhD thesis, Albert-Ludwigs-Universität Freiburg im Breisgau, 2008.
- [ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, editors, *Computer Security - ESORICS 2005*, volume 3679 of *Lecture Notes in Computer Science*, pages 159–177. Springer, 2005.
- [ADFM06] Giuseppe Ateniese, Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci. Provably-Secure Time-Bound Hierarchical Key Assignment Schemes. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 288–297. ACM, 2006.
- [AF96] Masayuki Abe and Eiichiro Fujisaki. How to date blind signatures. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1996.
- [AFB05] Mikhail J. Atallah, Keith B. Frikken, and Marina Blanton. Dynamic and Efficient Key Management for Access Hierarchies. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security*, pages 190–202. ACM, 2005.
- [AIR01] Bill Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *In Birgit Pfitzmann, editor, Advances in Cryptology EUROCRYPT 2001, volume 2045 of Lecture Notes in Computer Science*, pages 119–135. Springer-Verlag, 2001.
- [ASW97] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *Proc. 4th ACM Conference on Computer and Communications Security*, pages 6–17, 1997.

- [ASW00] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):591–610, April 2000.
- [BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 566–582. Springer-Verlag, 2001.
- [BCC04] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proc. 11th ACM Conference on Computer and Communications Security*, pages 225–234. acm press, 2004.
- [BCGS09] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard Java Card. In to appear, editor, *ACM Conference on Computer and Communications Security*, 2009.
- [BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.
- [BDD07] Stefan Brands, Liesje Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 400–415. Springer, 2007.
- [BDLS10] M. Bezzi, S. De Capitani di Vimercati, G. Livraga, and P. Samarati. Protecting privacy of sensitive value distributions in data release. In *Proc. of the 6th International Workshop on Security and Trust Management (STM 2010)*, Athens, Greece, September 2010.
- [Bez07] M. Bezzi. An entropy-based method for measuring anonymity. In *Proc. of the 3rd International Workshop on the Value of Security through Collaboration (SECOVAL 2007)*, Nice, France, 2007.
- [Bez10] M. Bezzi. Expressing privacy metrics as one-symbol information. In *Proc. of the 2010 EDBT/ICDT Workshops*, Lausanne, Switzerland, 2010.
- [BF03] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
- [BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer Verlag, 2003.
- [BP10] Stefan Brands and Christian Paquin. E-participation proof of concept with u-prove technology. online, 2010. Available from <http://www.microsoft.com/mscorp/twc/endoendtrust/vision/eid.aspx>, accessed in November 2010.
- [BR01] Vincent Buskens and Werner Raub. Embedded trust: Control and learning. In Ed Lawler and Shane Thye, editors, *Group Cohesion, Trust, and Solidarity*, volume 19 of *Advances in Group Processes*, pages 167–202, 2001.
- [Bra99] Stefan Brands. *Rethinking Public Key Infrastructure and Digital Certificates—Building in Privacy*. PhD thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands, 1999.
- [BS04] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick McDaniel, editors, *ACM CCS 2004*, pages 168–177. ACM, 2004.
- [BSZ05] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.

- [Byg02] L. Bygrave. *Data Protection Law, Approaching Its Rationale, Logic and Limits*. Kluwer Law International, The Hague, London, New York, 2002.
- [Cam06] Jan Camenisch. Protecting (anonymous) credentials with the trusted computing group's tpm v1.2. In Simone Fischer-Hübner, Kai Rannenberg, Louise Yngström, and Stefan Lindskog, editors, *SEC*, volume 201 of *IFIP*, pages 135–147. Springer, 2006.
- [CD00] Jan Camenisch and Ivan Damgård. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer Verlag, 2000.
- [CDF<sup>+</sup>07] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation and encryption to enforce privacy in data storage. In *Proc. of the 12th European Symposium On Research In Computer Security (ESORICS 2007)*, Dresden, Germany, September 2007.
- [CDF<sup>+</sup>09a] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation design for efficient query execution over sensitive distributed databases. In *Proc. of the 29th International Conference on Distributed Computing Systems (ICDCS 2009)*, Montreal, Quebec, Canada, June 2009.
- [CDF<sup>+</sup>09b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Keep a few: Outsourcing data while maintaining confidentiality. In *Proc. of the 14th European Symposium On Research In Computer Security (ESORICS 2009)*, Saint Malo, France, September 2009.
- [CDF<sup>+</sup>10] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM Transactions on Information and System Security (TISSEC)*, 13(3):1–33, July 2010.
- [CDN09] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access control. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 131–140. ACM, 2009.
- [CDN10] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Unlinkable priced oblivious transfer with rechargeable wallets. In *Financial Cryptography 2010*, 2010.
- [CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer Verlag, 1994.
- [CF85] Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS 1985*, pages 372–382. IEEE, 1985.
- [CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 319–327. Springer, 1988.
- [CGP<sup>+</sup>08] Stelvio Cimato, Marco Gamassi, Vincenzo Piuri, Roberto Sassi, and Fabio Scotti. Privacy-Aware Biometrics: Design and Implementation of a Multimodal Verification System. In *ACSAC*, pages 130–139. IEEE Computer Society, 2008.
- [Cha81] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.
- [Cha83] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology — Proceedings of CRYPTO '82*, pages 199–203. Plenum Press, 1983.
- [Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.
- [Cha88] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

- [CHK<sup>+</sup>06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *ACM Conference on Computer and Communications Security*, pages 201–210, 2006.
- [CHL06] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Balancing accountability and privacy using e-cash (extended abstract). In *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 141–155, 2006.
- [CK01] Sebastian Clauß and Marit Köhntopp. Identity management and its support of multi-lateral security. *Computer Networks*, 37(2):205–219, October 2001.
- [CKS09] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, pages 481–500, 2009.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.
- [CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer Verlag, 2002.
- [CL05] Jan Camenisch and Anna Lysyanskaya. A formal treatment of onion routing. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 169–187. Springer Verlag, 2005.
- [CMN<sup>+</sup>10] Jan Camenisch, Sebastian Mödersheim, Gregory Neven, Franz-Stefan Preiss, and Dieter Sommer. A language enabling privacy-preserving access control. In *To appear at SACMAT 2010*. ACM, 2010.
- [CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In Moni Naor, editor, *Eurocrypt 2007*, *Lecture Notes in Computer Science*, pages 573–590, 2007.
- [CPHH02] Sebastian Clauß, Andreas Pfitzmann, Marit Hansen, and Els Van Herreweghen. Privacy-enhancing identity management. *The IPTS Report*, 67:8–16, September 2002.
- [CS03] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144, 2003.
- [CSF<sup>+</sup>08] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008.
- [CvH91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer-Verlag, 1991.
- [CW88] C. Camerer and K. Weigelt. Experimental tests of a sequential equilibrium reputation model. *Econometrica*, 56:1–36, 1988.
- [Das00] Partha Dasgupta. Trust as a commodity. In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Department of Sociology, University Oxford, 2000.
- [Del00] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 150–157, New York, NY, USA, 2000. ACM Press.
- [Del03] Chrysanthos Dellarocas. The digitization of word-of-mouth: Promise and challenges of online feedback mechanisms. *Management Science*, pages 1407–1424, October 2003.

- [Del06] Chrysanthos Dellarocas. Research note – how often should reputation mechanisms update a trader’s reputation profile? *Information Systems Research*, 17(3):271–285, 2006.
- [DFJ<sup>+</sup>07] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Over-encryption: Management of Access Control Evolution on Outsourced Data. In Christoph Koch, Johannes Gehrke, Minos N. Garofalakis, Divesh Srivastava, Karl Aberer, Anand Deshpande, Daniela Florescu, Chee Yong Chan, Venkatesh Ganti, Carl-Christian Kanne, Wolfgang Klas, and Erich J. Neuhold, editors, *VLDB*, pages 123–134. ACM, 2007.
- [DFJ<sup>+</sup>08] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati. Preserving Confidentiality of Security Policies in Data Outsourcing. In Vijay Atluri and Marianne Winslett, editors, *WPES*, pages 75–84. ACM, 2008.
- [DFJ<sup>+</sup>10a] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragments and loose associations: Respecting privacy in data publishing. *Proc. of the VLDB Endowment*, 3(1):1370–1381, 2010.
- [DFJ<sup>+</sup>10b] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, Gerardo Pelosi, and Pierangela Samarati. Encryption-based Policy Enforcement for Cloud Storage. In *Proceedings of the 1st ICDCS Workshop on Security and Privacy in Cloud Computing (SPCC’10)*, Genova, Italy, Jun. 2010.
- [DFJ<sup>+</sup>10c] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Encryption Policies for Regulating Access to Outsourced Data. *ACM Trans. Database Syst.*, 35(2), 2010.
- [DFPS07] S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, and P. Samarati. Privacy of outsourced data. In A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. De Capitani di Vimercati, editors, *Digital Privacy: Theory, Technologies and Practices*. Auerbach Publications (Taylor and Francis Group), 2007.
- [DFRM09] J. Domingo-Ferrer and D. Rebollo-Monedero. Measuring Risk and Utility of Anonymized Data Using Information Theory. In *Proc. of the 2009 EDBT/ICDT Workshops*, Saint-Petersburg, Russia, 2009.
- [DKD<sup>+</sup>09] Claudia Diaz, Eleni Kosta, Hannelore Dekeyser, Markulf Kohlweiss, and Girma Nigussie. Privacy preserving electronic petitions. *Identity in the Information Society*, 1(1):203–209, 2009.
- [DM99] M. R. DeWeese and M. Meister. How to measure the information gained from one symbol. *Network: Comput. Neural Syst*, 10:325–340, 1999.
- [DMS04] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. TOR: The Second-Generation Onion Router. In *USENIX Security Symposium*, pages 303–320. USENIX, 2004.
- [Dou02] John R. Douceur. The sybil attack. In *IPTPS ’01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.
- [ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology — CRYPTO ’84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer Verlag, 1985.
- [ENI07] ENISA. Position paper. reputation-based systems: a security analysis, 2007.
- [EP10] Anders Ellvin and Tobias Pulls. Implementing a Privacy-Friendly Secure Logging Module into the PRIME Core. Master thesis, Department of Computer Science, Karlstad University, Sweden, 2010.
- [Fan61] R. M. Fano. *Transmission of Information; A Statistical Theory of Communications*. MIT University Press, New York, NY, USA, 1961.
- [FOO91] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. Interactive bi-proof systems and undeniable signature schemes. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT ’91*, volume 547 of *Lecture Notes in Computer Science*, pages 243–256. Springer-Verlag, 1991.



- [FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology - AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992.
- [FR99] Eric Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, August 1999.
- [FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1987.
- [FWCY10] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 42(4):1–53, 2010.
- [Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 169–178. ACM, 2009.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability; a Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.
- [GLM<sup>+</sup>04] Marco Gamassi, Massimo Lazzaroni, Mauro Nicola Misino, Vincenzo Piuri, Daniele Sana, and Fabio Scotti. Accuracy and performance of biometric systems. In *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference, 2004. IMTC 04.*, volume 1, pages 510–515 Vol.1, May 2004. 0-7803-8248-X.
- [GPSS05] Marco Gamassi, Vincenzo Piuri, Daniele Sana, and Fabio Scotti. Robust Fingerprint Detection for Access Control. In *Proceedings of the Second RoboCare Workshop, Rome, Italy, May, 2005*. Italian National Research Council (CNR), 2005.
- [GRS99] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):84–88, February 1999.
- [Hed09] Hans Hedbom. A survey on transparency tools for privacy purposes. In Vashek Matyas, Simone Fischer-Hübner, Daniel Cvrcek, and Petr Svenda, editors, *The Future of Identity in the Information Society*, volume 298 of *IFIP Advances in Information and Communication Technology*. Springer Boston, 2009.
- [HH10] Tobias Pulls and Hans Hedbom. Unlinking database entries: Implementation issues in privacy preserving secure logging. In *2nd International Workshop on Security and Communication Networks (IWSCN)*, 2010.
- [Hil09] Mireille Hildebrant. D 7.12: Biometric behavioural profiling and transparency enhancing tools. FIDIS WP7 Deliverable, March 2009.
- [HIM02a] H. Hacigümüş, B. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of 18th International Conference on Data Engineering (ICDE 2002)*, San Jose, California, USA, February 2002.
- [HIM02b] H. Hacigümüş, B. Iyer, and S. Mehrotra. Providing database as a service. In *Proc. of 18th International Conference on Data Engineering (ICDE 2002)*, San Jose, California, USA, February 2002.
- [Hol06] J. E. Holt. Logcrypt: forward security and public verification for secure audit logs. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research - Volume 54*, volume 167 of *ACM International Conference Proceeding Series*, pages 203–211. Australian Computer Society, 2006.
- [HPL10] Hans Hedbom, Tobias Pulls, Peter Hjältquist, and Andreas Laven. Adding Secure Transparency Logging to the PRIME Core. In Michele Bezzi, Penny Duquenoy, Simone Fischer-Hübner, Marit Hansen, and Ge Zhang, editors, *Privacy and Identity Management for Life*, volume 320 of *IFIP Advances in Information and Communication Technology*. Springer Boston, 2010.
- [ISO] ISO/IEC 18033-2: 2006: Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers.

- [jav10] Java card technology. online, 2010. Available from <http://www.oracle.com/technetwork/java/javacard/overview/index.html>, accessed in November 2010.
- [JMSW02] Robert Johnson, David Molnar, Dawn Xiaodong Song, and David Wagner. Homomorphic signature schemes. In Bart Preneel, editor, *Topics in Cryptology - CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 244–262. Springer, 2002.
- [Ker09] Florian Kerschbaum. A verifiable, centralized, coercion-free reputation system. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 61–70, New York, NY, USA, 2009. ACM.
- [KK09] Sandeep S. Kumar and Paul Koster. Portable reputation: Proving ownership across portals. In *Proc. of the European Context Awareness and Trust 2009 (EuroCAT09), 3rd Workshop on Combining Context with Trust, Security, and Privacy*, volume 504, pages 21–30. CEUR Workshop Proceedings, September 2009.
- [LL09] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, Paris, France, 2009.
- [LLV07] N. Li, T. Li, and S. Venkatasubramanian.  $t$ -closeness: Privacy beyond  $k$ -anonymity and  $\ell$ -diversity. In *Proc. of the 23rd IEEE International Conference on Data Engineering (ICDE 2007)*, Istanbul, Turkey, 2007.
- [LRSW99] Anna Lysyanskaya, Ron Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard Heys and Carlisle Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.
- [Mau96] Ueli Maurer. Modelling a public-key infrastructure. In E. Bertino, editor, *European Symposium on Research in Computer Security — ESORICS '96*, volume 1146 of *Lecture Notes in Computer Science*, pages 325–350. Springer-Verlag, September 1996.
- [MGK06] A. Machanavajjhala, J. Gehrke, and D. Kifer.  $\ell$ -diversity: Privacy beyond  $k$ -anonymity. In *Proc. of the 22nd International Conference on Data Engineering (ICDE '06)*, Atlanta, GA, April 2006.
- [MO04] Tobias Mahler and Thomas Olsen. Reputation systems and data protection law. In *eAdoption and the Knowledge Economy: Issues, Applications, Case Studies*, pages 180–187, Amsterdam, 2004. IOS Press.
- [MT07] Di Ma and G. Tsudik. Forward-secure sequential aggregate authentication. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 95–100. IEEE, 2007.
- [MT08] Di Ma and G. Tsudik. A new approach to secure logging. In *Data and Applications Security XXII. 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, 2008.
- [Mui03] Lik Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD Thesis, Massachusetts Institute of Technology, 2003.
- [NFHF09] Toru Nakanishi, Hiroki Fujii, Yuta Hira, and Nobuo Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In Stanislaw Jarecki and Gene Tsudik, editors, *Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 463–480. Springer, 2009.
- [NP99] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.
- [OS07] Rafail Ostrovsky and William E. Skeith III. Private searching on streaming data. *Journal of Cryptology*, 20(4):397–430, 2007.
- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–239. Springer Verlag, 1999.
- [pria] PrimeLife project, website. [www.primelife.eu](http://www.primelife.eu).
- [PR1b] PRIME. Privacy and Identity Management for Europe. <http://www.PRIME-project.eu/>.



- [PRT04] Elan Pavlov, Jeffrey S. Rosenschein, and Zvi Topol. Supporting privacy in decentralized additive reputation systems. In *The Second International Conference on Trust Management*, pages 108–119, Oxford, United Kingdom, March 2004.
- [PS08] Franziska Pingel and Sandra Steinbrecher. Multilateral secure cross-community reputation systems. In S.M. Furnell S.K. Katsikas and A. Lioy, editors, *Proceedings of Trust and Privacy in Digital Business, Fifth International Conference, TrustBus*, volume 5185 of *Lecture Notes in Computer Science*, pages 69–78. Springer, 2008.
- [Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.
- [RKP09] Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally composable adaptive priced oblivious transfer. In *Pairing '09: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography*, pages 231–247, Berlin, Heidelberg, 2009. Springer-Verlag.
- [RKZF00] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.
- [RMFDF09] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer. From  $t$ -closeness-like privacy to postrandomization via information theory. *IEEE Transactions on Knowledge and Data Engineering*, 22(11):1623–1636, 2009.
- [RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.
- [Sam01] P. Samarati. Protecting respondents’ identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, November 2001.
- [SCS10] Stefan Schiffner, Sebastian Clauß, and Sandra Steinbrecher. Privacy and liveliness for reputation systems. In *Proceedings of 2009 European PKI Workshop (EuroPKI'09)*, volume 6391 of *Lecture Notes in Computer Science*. Springer, 2010.
- [SCS11] Stefan Schiffner, Sebastian Clauß, and Sandra Steinbrecher. Privacy, liveliness and fairness for reputation. In *Proceedings of 37th Conference on Current Trends in Theory and Practice of Computer Science*, volume 6543 of *Lecture Notes in Computer Science*. Springer, 2011.
- [SGM09] S. Steinbrecher, S. Groß, and M. Meichau. Jason: A scalable reputation system for the semantic web. In *Proceedings of IFIP Sec 2009, IFIP International Information Security Conference: Emerging Challenges for Security, Privacy and Trust*, volume 297 of *IFIP AICT*, pages 421–431. Springer, May 2009.
- [SK98] B. Schneier and J. Kelsey. Cryptographic support for secure logs on untrusted machines. In *The Seventh USENIX Security Symposium Proceedings*, pages 53–62. USENIX Press, January 1998.
- [SSA06] S. Sackmann, J. Strüker, and R. Accorsi. Personalization in privacy-aware highly dynamic systems. *COMMUNICATIONS OF THE ACM*, 49(9), September 2006.
- [Ste06] Sandra Steinbrecher. Design options for privacy-respecting reputation systems within centralised internet communities. In *Proceedings of IFIP Sec 2006, 21st IFIP International Information Security Conference: Security and Privacy in Dynamic Environments*, volume 201 of *IFIP*, pages 123–134. Springer, May 2006.
- [Ste09] Sandra Steinbrecher. Enhancing multilateral security in and by reputation systems. In *Proceedings of the IFIP/FIDIS Internet Security and Privacy Summer School, Masaryk University Brno, 1-7 September 2008*, volume 298 of *IFIP AICT*, pages 135–150. Springer, 2009.
- [Ste10] Sandra Steinbrecher. The need for interoperable reputation systems. In *Proceedings of IFIPiNetSec 2010 Open Research Problems in Network Security, Sofia, Bulgaria, 05-06 March 2010*. To be published by Springer, 2010.
- [SWP00] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

- [TBD<sup>+</sup>10] Carmela Troncoso, Josep Balasch, Claudia Diaz, Venelin Gornishki, Markulf Kohlweiss, Michal Sterckx, and Victor Sucasas. Adapid d15: E-government ii. Adapid deliverable, 2010.
- [Tob02] Christian Tobias. Practical oblivious transfer protocols. In Fabien A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop, IH 2002*, volume 2578 of *Lecture Notes in Computer Science*, pages 415–426. Springer, 2002.
- [VLV<sup>+</sup>08] Kristof Verslype, Jorn Lapon, Pieter Verhaeghe, Vincent Naessens, and Bart De Decker. PetAnon: A privacy-preserving e-petition system based on Idemix. CW Reports CW522, Department of Computer Science, K.U.Leuven, October 2008.
- [Vos04] Marco Voss. Privacy preserving online reputation systems. In *International Information Security Workshops*, pages 245–260. Kluwer, 2004.
- [W3C06] W3C. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification, 2006. <http://www.w3.org/TR/P3P11/>.
- [WABL<sup>+</sup>06] J. Weitzner, H. Abelson, T. Berners-Lee, C. Hanson, J. Hendler, L. Kagal, D. L. McGuiness, G. J. Sussman, and K. Waterman. Transparent accountable data mining: New strategies for privacy protection. Computer Science and Artificial Intelligence Laboratory Technical Report MIT-CSAIL-TR-2006-007, Massachusetts Institute of Technology, Cambridge, Ma, USA, 2006.
- [WSLP08] Karel Wouters, Koen Simoens, Danny Lathouwers, and Bart Preneel. Secure and privacy-friendly logging for e-government services. In *ARES*, pages 1091–1096. IEEE Computer Society, 2008.



# **Part III**

## **HCI**

## Introduction

PrimeLife has the vision of bringing sustainable and user-controlled Privacy and Identity Management to future networks and services. User-controlled Privacy and Identity Management implies that users can make informed choices about the selection of appropriate (anonymous) credentials for proving personal properties, the release of personal data items, as well as the selection and adaption of privacy and trust policies. For enabling well-informed decisions, the user interfaces have to present information about the trustworthiness of communication partners. Additionally, information about the privacy implications of those choices in terms of the linkability of the user's partial identities and of their communication partners' data handling policies should be displayed. This information about choices and their implications must be well understandable, noticeable and must comply with legal requirements. At the same time, it should not be perceived as too interfering or disturbing. Actions needed for performing choices should also be easy to handle. The challenges of researching and developing user interfaces for Privacy and Identity Management that are intuitive, user-friendly and compliant with legal and social requirements have been addressed by PrimeLife Activity 4 (HCI).

In this part of the book, we will present the HCI research within PrimeLife, which aims in particular at addressing the following research challenges:

- Available system usability scales and questionnaires for measuring user experiences and usability of various HCI aspects do not address PET related issues. A special focus of the research work of the HCI work within PrimeLife has therefore been on the development of novel methodologies for evaluating HCI solutions for PETs that can be used within PrimeLife. Chapter 10 of this part presents *PET-USES* (Privacy-Enhancing Technology Users Self-Estimation Scale), which we have developed and used within PrimeLife to evaluate PrimeLife user interfaces.
- Privacy-enhancing technologies (PETs) are based on complex technical concepts or constructs such as pseudonyms, unlinkability and anonymous credentials that are unfamiliar to many end users and often do not fit to their mental picture of what belongs to an electronic identity and how it can be technically protected. How can a notion about privacy and electronic identity be illustrated to the user for estimating the risk of being identified across different interactions with one or several communication partners? How can the user be assisted in understanding and taking advantage of the privacy-enhancing features of PrimeLife technologies? Chapter 13 on *HCI for PrimeLife Prototypes* presents the HCI work done in terms of development and testing of PrimeLife prototypes developed within the PrimeLife Activities 1 and 2 (described in Parts I and II of this book). In Chapter 12 (*The Users' Mental Models' Effect on their Comprehension of Anonymous Credentials*), we analyse what effects the users' mental models have on their understanding of the selective disclosure property of anonymous credentials.
- How can the user interfaces mediate reliable trust in *PrimeLife* technology and communication partners to end users? For addressing this problem, we have con-

ducted research on the design of a user-friendly trust evaluation function that can convey information stated or certified by third (trustworthy) parties to end users about the level of trustworthiness of communication partners. User-friendly transparency tools are also a means for enhancing the users' trust in PrimeLife technologies. Within PrimeLife, we have developed and tested the data track, which is a transparency tool that provides the user with a history function documenting what personal data the user has revealed under which conditions. The data track also includes online functions allowing a user to exercise her right to access her data on remote services sides. Chapter 13 on *Trust and Assurance HCI* reports on the HCI work for the trust evaluation function and PrimeLife data track.

- How can privacy and trust policy definitions, administration and negotiations be simplified for end users by appropriate means for policy presentation, predefined settings and automation in a way compliant with European privacy legislation? In PrimeLife, we have researched novel concepts for a simplified management of privacy preferences and user-friendly display of data handling policies of services sides including information about how closely they match the user's privacy preferences. Results of our research on policy-related HCI aspects are reported in Chapter 14 on *HCI for Policy Display and Administration* and Chapter 15 on *Privacy Policy Icons*.



# Chapter 10

## PET-USES

Erik Wästlund and Peter Wolkerstorfer

**Abstract** This chapter describes the PET-USES [Privacy-Enhancing Technology Users' Self-Estimation Scale], a questionnaire that enables users to evaluate PET User Interfaces [UIs] for their overall usability and to measure six different PET aspects. The objective of this chapter is to outline the creation and the background of the PET-USES questionnaire and invite the usability community to not only use the test but also contribute to the further development of the PET-USES. This text is an excerpt of [WWK10] which additionally contains a more elaborate description of the rationale behind the PET-USES.

### 10.1 Introduction

PET-USES [Privacy-Enhancing Technology Users' Self Estimation Scale] important thing is a questionnaire that enables users to evaluate PET-User Interfaces [UIs]. The reason for developing and using PET-USES was to be able to measure the perceived usability of UIs, both during single user trails and during large group walkthroughs of screen recordings. Although there are a number of questionnaires measuring user experience, usability and various HCI (human-computer interaction) aspects such as the hedonic quality [HBK03] of both software and websites [Bro96] [TS04], to our knowledge none include PET-related issues.

The PET-USES consists of two major parts of questions: one part measuring overall usability and one part measuring PET-aspects. Thus, the PET-usability scales have a dual purpose. They evaluate the software's general usability and the extent to which the software assists the user in learning and understanding privacy-related issues. The PET-USES questionnaire consists of the following modules (the detailed content can be seen in the Appendix):

Part I – Usability:

- General Usability
- Ease of Learning



- Ease of Use
- User Value

Part II – PET-related aspects:

- Data Management
- Credential Management
- PrivPrefs<sup>1</sup>
- Recipient Evaluation
- Data Release
- History

An important feature of the measurement of PET-aspects is the modularity of the questionnaire, enabling the inclusion or exclusion of scales measuring specific aspects based on the tasks and features being evaluated, e.g., dependent on the context of use, the Credential Management part could be excluded from the questionnaire.

The PET-USES questionnaire is based on the ISO 9241 general standard of usability [ISO88] as well as the more domain specific HCI guidelines presented by Patrick et al. 2003 [PKHvB03] and utilised in the work with the PRIME integrated IDM prototype [Pet05]. The former defines usability as the “extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction,” whereas the latter promotes the four categories comprehension (to understand or know), consciousness (be aware or informed), control (to manipulate or be empowered) and consent (to agree). Although the two views might seem divergent at first, they can readily be combined within the structure of usability testing proposed by Hornbæk [Hor06]. Based on a review of 180 studies, published in core HCI journals and proceedings, he argues for a change in terminology from the ISO 9241, to better encompass what is actually being measured.

Effectiveness and efficiency are often measured in a more objective fashion than the user self estimations of the PET-USES. The effectiveness of a given interface can for instance be measured in terms of task completion time and efficiency in terms of quality of task solution [FHH00] and, of course, optimally usability evaluations should be comprised of a combination of self estimation and more objective measurements. It should, however, be pointed out that these types of measurement require fully functional interfaces, whilst the PET-USES can be used in a much earlier stage to measure users’ perception as estimates of effectiveness and efficiency.

The PET-USES scale General Usability is measured as a composite of the subscales Ease of Learning, Ease of Use and User Value. The rationale for differentiating between the subscales Ease of Learning and Ease of Use is that intuitive interfaces are perceived to have a better learnability whereas a less intuitive interface can be used easily only once the user is accustomed to it. It is also noteworthy that the General Usability value will be less influenced by perceived User Value than

---

<sup>1</sup> PrivPrefs (Privacy Preferences) is a method that is currently being investigated in the PrimeLife project for defining personal privacy preferences which will be used for automated evaluations of the appropriateness of data-requests. The PrivPrefs are similar to the privacy preferences as defined in P3P ([www.w3.org/P3P](http://www.w3.org/P3P)).

Ease of Learning and Ease of Use. This reflects the fact that, although user value is an important driver for software adoption, the focus lies more on the usability of a product than its perceived benefits. The PET-aspects modules currently developed are: Data Management, Credential Management, PrivPrefs, Recipient Evaluation, Data Release, and History. They can all be used to evaluate specific PET-related functionality of software or websites. (See appendix for the entire PET-USES questionnaire.)

The focuses of the scales are the following privacy-critic areas:

- **Data Management:** The extent to which the system makes it easier to store and organise personal information. This scale can be used to evaluate all types of identity management software and services.
- **Credential Management:** The extent to which the system makes it easier to store and organise certificates and credentials. This scale can be used to evaluate identity management systems that include issued claim credentials (e.g., the Higgins project<sup>2</sup>).
- **PrivPrefs:** This scale is designed to measure the extent to which the system makes it easier to set general and excessive levels for data release policies and to what extent the user is informed of unwanted data dissemination. Thus, an aspect of this scale is the decision support qualities of the system.
- **Recipient Evaluation:** the extent to which the system helps users evaluate the data recipients' credibility and trustworthiness. This scale can also be regarded in terms of decision support.
- **Data Release:** The extent to which the system clarifies what personal information is being released and who is the recipient of the data.
- **History:** The extent to which the system can show the user when, what and, to whom personal information has been released and thus provide an overview of what data any given service provider might have accumulated.

## 10.2 PET-USES in Practice

So far, the PET-USES has been used in a few different settings for different purposes. However, even though it has been used both by researchers and commercial activities outside the PrimeLife, to the best of our knowledge the main usage has been to evaluate the PrimeLife prototypes. Based on our own experience and feedback from others, the modularity of the PET-USES is, much appreciated feature as this allows the test to be tailored to the fit the test scenario.

---

<sup>2</sup> [www.eclipse.org/higgins/](http://www.eclipse.org/higgins/)

### ***10.2.1 When to use the PET-USES***

The main reason for conducting usability tests is to discriminate between usable and unusable interfaces either during the design process or in comparisons between different systems. Typical use cases for the PET-USES include both of these scenarios. The PET-USES can be used both to compare the perceived usability strengths and weaknesses between different interfaces, and also to aid interface designers during the design process. The latter by administrating the test at various steps in the process. However, as with all statistical testing, the possibility of finding significant results is dependent on the a-priori power of the investigation. When it comes to comparing existing interfaces, a bigger effects size can be achieved both by choosing and comparing interfaces that are very good with interfaces that are very bad or by enrolling more users into the test. During interface design, especially during fast iterations, the differences between versions are usually quite small and the tested user group rather small and hence the power of a test such as the PET-USES will become quite small. This should be taken into consideration when planning when to use the PET-USES, as it will be more useful when evaluating clear steps in the design process. In order to gain power by adding more respondents without having to do a great number of complete user tests, it is possible to do large group walkthroughs of screen recordings. An additional feature of this method is that it is possible to do user tests on interfaces without any functionality.

### ***10.2.2 How to use the PET-USES***

In order to facilitate both the use and the evaluation of the PET-USES, a web service has been set up at CURE.<sup>3</sup> The web service enables research companies to use the PET-USES questionnaire for their evaluations and will be open to all who wish to use the PET-USES on the premises that the collected PET-USES data will be used to gather feedback and further develop the questionnaire and its scales. In addition to using the scales of the PET-USES, researchers in this area will have the possibility to suggest new modules for inclusion in the sub-scale battery to reflect the ever changing field of PETs. Data provided on the website will be anonymised and treated confidentially. Only those conducting the research and the creators of the PET-USES (i.e., Karlstad University and CURE) will have access to the data provided. Users of the site who wish to retain data from other sources than the PET-USES are of course allowed to do so, but in order to evaluate the PET-USES, users are encouraged to provide data that can facilitate the validation of the test.

---

<sup>3</sup> <http://pet-uses.cure.at>

## 10.3 Conclusions

The PET-USES test presented in this chapter is a usability questionnaire that focuses on measures of aspects of General Usability as well as specifically tailored scales that measure the usability of PET solutions. The test is grounded in current views on usability and the experience so far of using the test show that both practitioners and users report that the PET-USES is an easy to use and informative tool. The CURE web service for using the PET-USES is open to PET researchers who wish to evaluate PET UIs. Our hope is that it will be a part of the PET development and evaluation toolkit used by usability researchers interested in the area of PETs.

## 10.4 Appendix: PET-USES[1.0]

Note: the headings and numerals in the following test are mainly for presentational purposes and thus optional during the use of PET-USES. Items 2, 3, 7, 8, and 21 should be reversed before summated.

### 10.4.1 Instructions

This test is designed to measure your experience with the system you've tested today. Your answers will be used to evaluate the system so please answer the questions as truthfully as you can. As the questions are designed to measure various aspects of the systems usability there are no right or wrong answers. Please use the scale below to indicate to what extent you disagree or agree to the statements that follow.

- 1 Strongly disagree
- 2 Disagree
- 3 Neither agree nor disagree
- 4 Agree
- 5 Strongly agree

**General usability**

1. I found it easy to learn how to use the *system*. 1 2 3 4 5
2. I had to learn a lot in order to use the *system*. 1 2 3 4 5
3. I keep forgetting how to do things with this *system*. 1 2 3 4 5
4. I need a lot of assistance to use this *system*. 1 2 3 4 5
5. I find the *system* interface easy to use. 1 2 3 4 5
6. I find the organisation of the *system* interface understandable. 1 2 3 4 5
7. I get confused by the *system* interface. 1 2 3 4 5
8. I find it very difficult to work with the *system*. 1 2 3 4 5
9. I find that the benefits of using the *system* are bigger than the effort of using it. 1 2 3 4 5
10. I would like to use this *system* regularly. 1 2 3 4 5

**Data management**

11. I get a clear view of my personal *data* from the *system*. 1 2 3 4 5
12. I find organising my personal *data* easy with this *system*. 1 2 3 4 5
13. I find keeping track of various user names and passwords is easy with this *system*. 1 2 3 4 5

**Credential management**

14. I find it easy to add personally issued credentials into the *system*. 1 2 3 4 5
15. I find it easy to add / import certificates into the *system*. 1 2 3 4 5
16. I find it easy to manage my certificates and credentials. 1 2 3 4 5

**PrivPrefs**

17. I find it easy to use settings for how much or how little *data* to be released. 1 2 3 4 5
18. I find that the *system* helps me understand the effects of different privacy settings. 1 2 3 4 5
19. I feel safer knowing that I will be notified if I'm about to release more *data* than my chosen preference. 1 2 3 4 5

**Recipient Evaluation**

20. The *system* makes it easy to decide if it is safe to release my *data*. 1 2 3 4 5
21. I don't understand how the *system* determines if a *data* recipient is trustworthy. 1 2 3 4 5
22. I feel safer releasing my personal *data* when the *system* states it's ok. 1 2 3 4 5

**Data Release**

23. I know what personal information I'm releasing. 1 2 3 4 5

24. I find it easy to decide how much or how little *data* to release in a given transaction. 1 2 3 4 5

25. I get help from the system to understand who will receive my *data*. 1 2 3 4 5

**History**

26. I can easily find out who has received my personal *data* with this *system*. 1 2 3 4 5

27. I get a good view of who knows what about me from this *system*. 1 2 3 4 5

28. I can easily see how much I've used a particular username with this *system*. 1 2 3 4 5



# Chapter 11

## HCI for PrimeLife Prototypes

Cornelia Graf, Peter Wolkerstorfer, Christina Hochleitner, Erik Wästlund, and Manfred Tscheligi

**Abstract** User-centered design (UCD) processes need to be further extended to the field of privacy enhancing technologies (PETs). The goal of the UCD process for PETs is to provide a means for users to empower them to manage their privacy on the Web. Taking care of privacy and being careful while surfing the Web are still considered to be cumbersome and time-consuming activities. Hence, PrimeLife aspires to provide easy to use tools for users to manage their privacy. This chapter describes the challenges in UCD that arose during the development of the PrimeLife prototypes. As part of the HCI activities in the PrimeLife project, we have researched the users' attitudes towards privacy and discovered the main challenges when developing user-friendly PETs. We use two example prototypes to explain how the challenges can be tackled in practice. In general, PETs should neither require much of the user's attention and time, nor should they require particular technical knowledge. They should, in fact, present the complex methods of privacy enhancing technologies in an easy, understandable and usable way. We will conclude this chapter with a discussion of our findings and implications for further development of user-centered privacy enhancing technologies.

### 11.1 Introduction

One of the main goals of the HCI activities within the PrimeLife project was the design, development and evaluation of usable and understandable privacy enhancing technology (PET) prototypes. This also included the extension of user-centered design (UCD) processes to be able to advance existing methods in order to be applicable to the particular needs and challenges of PET prototypes. The PrimeLife prototypes were also developed to be able to answer research questions by conducting user evaluations. As a part of this process, researchers encountered several challenges to be solved in order to accomplish the above stated goal. In the present chapter, these challenges will be identified in Section 11.2 and a brief outline of



them will be provided. The main goal of this chapter is not an in-depth description of the challenges, but to provide an overview of possible issues and solutions when developing PETs. Section 11.3 will present examples on how these challenges were tackled as part of the PrimeLife project. Different approaches to apply the suggested solutions to the challenges within a UCD process will be described in Section 11.4. The findings presented in this chapter will be discussed in Section 11.5. Furthermore, conclusions will be drawn and an outlook on further research will be provided.

## 11.2 Overview of HCI challenges

In this section we will outline the HCI and UCD challenges we identified while working on the design and evaluation of various PrimeLife prototypes.

### *11.2.1 Challenge 1: Limited User Knowledge of PETs*

When designing and developing standard software, developers usually rely on knowledge in the form of research and products that already exist. Another possible source of information is the user's mind [Nor88], i.e., the user's knowledge about PETs and the application of these technologies.

Knowledge about privacy enhancing technologies in the mind of the users is still fairly limited. Hence, relying on this knowledge when design decisions are made in the area of PETs is likely to lead to unusable results. Several evaluations conducted throughout the duration of the PrimeLife project have indicated that the users' knowledge of privacy on the Web is rising. We see this as a consequence of more public occurrences and lively discussions about privacy in mass media, especially in connection to data disclosure and social networks [GA05]. Our experience gained during the PrimeLife project shows that an increasing number of users are interested in privacy and in active privacy protection.

Unfortunately most users still think that privacy protection is very time-consuming, too complicated or cannot be achieved, as it is in the hand of the service providers. Through several user evaluations within the last three years [KWW08, KWGT09, GWKT10], we have observed an increase in user awareness for privacy enhancing technologies and privacy issues in general. This might also be caused by added media coverage and attention to social networks or larger cases of industrial data loss. Concerning the users, this means that the knowledge in the mind of the users is increasing and based on our research it is foreseeable that general applicable guidelines for the best way of designing PETs will evolve over the next years.

### ***11.2.2 Challenge 2: Technologically Driven Development of PETs***

Currently PETs are mainly developed from a technological viewpoint [WT99, SBHK06]. It seems to lie in the nature of new and rising domains in software development that they often emerge from a rather technological viewpoint. Thus, they are targeted for technologically-minded expert users that are skilled in the use of complex interfaces, offering a vast number of interaction possibilities and options. When a domain is established, users as well as developers care about usability and user experience (the user's perception of the system)—unfortunately this is not enough. HCI engineers know that users should be included as early as possible in a project – a fact that is not widely spread in the creation of software domains [Iiv04]. As a result, unusable software is created within these novel areas. For understanding and accessing this fast growing selection for PETs and the background of PETs, technical knowledge from the users' side is required. This renders a lot of PETs unusable for most users. Since PETs are part of the aforementioned novel software domains, the evolution from technologically driven development towards user-centered development poses a major challenge.

### ***11.2.3 Challenge 3: Understanding PET Related Terms***

In order to use software successfully, users have to understand the meaning of text and labels that appear in the user interfaces. Research throughout the first of three project years of the PrimeLife project has indicated that users have problems understanding privacy-related terms and privacy policies. Several other studies also showed that the language and format of privacy policies are hardly understandable for most users [CGA06, FBL04, Rod03, SSM10]. There are different approaches to overcome the problem of policy understanding. Kelley presented a way to display websites' privacy policies in a more user friendly way [KBCR09]. Our studies confirm these findings for complex terms (e.g., "privacy preference") but we have also seen that basic terms, such as "privacy policy," are commonly understood. The study was conducted as an online evaluation where 73 volunteers participated. The participants defined their understanding of terms and rated the level of understanding. In a second step, the accuracy of these definitions was rated by security and privacy experts. Our results showed that users are able to understand most privacy terms, even privacy terms out of context were understandable for most users.

### ***11.2.4 Challenge 4: Wrong Mental Models of PETs***

Mental models describe how users expect things to work. Hence, they are the basis for interaction decisions. Various evaluations of PrimeLife prototypes demonstrated that users on the one hand have an incorrect mental model about the meaning and

importance of privacy. On the other hand, incorrect mental models of the Internet itself hinder the development of working mental models about private data on the Web. Mental model research is a very important method extension to research on the users' ideas, behaviour and assumptions about PETs. To increase our knowledge in this area, we conducted a mental model study with 17 participants, where we investigated users' ideas on how data handling and data storage on the Web may work. Our results showed that the users' understanding on this topic is incorrect and that there is much need for explaining how data handling and data storage works in reality. Figure 11.1 shows an example of the users' understanding of how data travels through the Internet. In this case, the user believes that the information he is accessing is transmitted via Microsoft from the original source.



Fig. 11.1: Participants' assumption on how data travels through the Web.

### 11.2.5 Challenge 5: Privacy as a Secondary Task

Managing privacy is not a primary task for users. Buying a book online is a primary task, while managing privacy while doing so is a secondary task. Users focus their attention onto the primary task; everything that is secondary does not get similar cognitive resources. Privacy protection is a support action for users when dealing with their primary task on the web. Consequently, as it often gets in the way of the users, they tend to ignore PETs in general [SF05]. In the PrimeLife project, similar to the security domain (where no user sits down in the evening with the idea to manage firewall settings as a free-time activity), we have seen that the amount of resources put into privacy management must be in due proportion to the task to be completed. Participants in PrimeLife evaluations often stated that installing and personalising a tool for privacy protection should not take longer than five to ten minutes.

### ***11.2.6 Challenge 6: Complex Mechanisms are Hard to Understand***

PETs are not easy to understand because they are based on complex models and therefore on complex background mechanisms, such as public-key encryption [WT99] or anonymous credentials. These mechanisms are difficult to communicate and explain to users due to a lack of technical knowledge and experience. Aside from manuals (which are known not to be read by users [NW06]), the user interface is the only possibility to communicate these concepts to the user. When talking about complexity and private data on the Web, we have to take into account that multiple sources of complexity are involved. The PET-related concepts (e.g., anonymous credentials) also increase the complexity experienced by the user.

## **11.3 Tackling the Challenges**

Together with recent discussions about privacy issues in mass media [GA05], users have become more aware of the concept of privacy and PETs in general. Nevertheless, privacy can still not be considered to be of particular importance to the wider public. Therefore, awareness training by projects such as PrimeLife, but also by public bodies is of particular importance in counteracting the current status. To empower the users to take care of their privacy, the challenges outlined in Section 11.2 need to be addressed. The following sections provide an overview of how the challenges were answered as part of the PrimeLife project. The approaches in the following are a general description of methods, the applied methods are described in the form of two example prototypes in Section 11.4.

### ***11.3.1 Limited User Knowledge of PETs***

In terms of interface design, we have discovered that the design aids the user in understanding PETs and privacy concepts, when they are offered with clear interfaces and structures that display privacy aspects and possible threats in an understandable way, as described in the Pattern collection developed for the PrimeLife interfaces [Pri10a]. Furthermore, the use of interfaces, not only by expert users, but also by less experienced persons, can be facilitated by providing two different views in the interface: a standard view for novice and average users that provides basic settings and does not need much configuration from the users' side, and an expert view that is especially designed for technologically minded persons, who set up and configure their own preferences. This principle was also applied in the creation of the backup prototype described in Section 11.4.1. Here, separate views for novice users and experts facilitate the understanding of privacy concepts and allow for more intuitive interaction and fine-grained control.

### ***11.3.2 Technologically Driven Development of PETs***

Based on our research results within the PrimeLife project, we believe that increasing the users' knowledge about privacy also eases their understanding of privacy concepts. Results of our research on user understanding and mental models of PETs can be found in [GKW<sup>+</sup>10, GWKT10, PWG10]. Privacy concepts can be easily conveyed through the user interface. Again, clear structures and visual aids, such as icons, can assist in understanding the technology. The interfaces have to be self-explanatory and usable by novice users through the employment of already known concepts (e.g., nutrition labels as privacy indicators [KBCR09]). Furthermore, privacy patterns [Pri10a, GWGT10] developed as a part of the PrimeLife project provide a basis for the creation of user-friendly PET interfaces.

As mentioned in Section 11.3.1, it is possible to create different approaches to the technology for different user groups. Thus, novice and less experienced users would need a rather simple interface requiring less technological knowledge that is not overwhelming with too much information and too many possibilities. In contrast to novice users, more experienced persons would need an expert view to access the advanced interaction possibilities to have full control of privacy settings.

### ***11.3.3 Understanding of PET Related Terms***

Besides increasing the users' knowledge of privacy and PET in general (cf. Section 11.3.1), it is utterly important that the employed wording is also understood by the users. Throughout several evaluations of the users' perception of privacy, security and PETs, as well as their understanding of connected processes, have been investigated [GKW<sup>+</sup>10, GWKT10]. This research has underlined the need for understandable wording and further explanation of unknown words. Terms should neither require a university degree in law nor in the field of security and privacy. [KBCR09] presented a way to display website privacy policies in a more user friendly way by creating information design that improved the comprehensibility and visual presentation of privacy policies. This was done by drawing from nutrition, warning, and energy labelling and considered to demonstrate that, compared to existing privacy policies, the new proposed privacy label allowed participants to find information more accurately and quickly, as well as providing a more enjoyable experience. A similar approach was used to create understandable and intuitive icons for use within the PrimeLife project [HHN10].

Research throughout the PrimeLife project has indicated that a quick evaluation of terms with only a few (non-expert) users can lead to indications on which words should be avoided in interfaces [KWGT09]. As part of the PrimeLife HCI activities, we have investigated several privacy terms that were also used in the created prototypes [GWKT10].

We identified the following five terms as being easiest to understand:

- Privacy protection
- Required data
- Digital traces
- Identity management
- Full privacy policy

The terms rated as being very hard to understand are the following:

- Anonymous credentials
- Privacy preference
- Linkability
- Privacy enhancing

### ***11.3.4 Wrong Mental Models of PETs***

A strong, user-centered design process can facilitate research on users' mental models as well as counter-activities to correct false assumptions. As part of the HCI Activities within PrimeLife, research on users' mental models was conducted [GKW<sup>+</sup>10] and applied to the developed prototypes. Through a well-founded requirement-gathering process, it is possible to form a picture of the users' understanding on how privacy and PETs work. Although being important in any user-centered process, the gathering of the users' mental models becomes even more important in novice domains such as PETs. It has to be ensured that the mental models are respected in the user interfaces. Furthermore, it is important to periodically review and update the researched mental models. This is necessary since privacy, as part of the ICT domain, is prone to fast development and rapid changes.

As part of the user-centered design approach, users from the target group are involved in the design process and therefore misunderstandings and false assumptions are discovered and can be clarified. It helps to provide mechanisms that assist users in understanding how the developed tool works. Furthermore, practical examples assist in correcting false assumptions. For the creation of all prototypes within PrimeLife it was taken care to consider user input as well as user feedback in various stages of the development process. An exemplary focus on users was particularly adhered to for the development of the backup prototype [Pri10b] (Section 11.4.1).

### ***11.3.5 Privacy as a Secondary Task***

Since privacy activities are considered to be cumbersome, the most efficient way to solve this challenge is by creating interfaces and tools that do not need much time and effort from the user side (see privacy patterns [Pri10a]). The PETs should work in the background: users should be able to access them easily, they should provide

support only when necessary and not disturb the users in their workflow with unnecessary pop-ups. The ease of use of interfaces in the complex setting of PETs was a central point in the PrimeLife project and was also applied to the developed prototypes [Pri10b].

### ***11.3.6 Complex Mechanisms are Hard to Understand***

Even for advanced users, privacy is a very complex concept. Furthermore, the employed interfaces usually try to convey very complex mechanisms in the background [KWW08]. Most users are overwhelmed with the kinds and amounts of information provided. Thus, it is important to get a clear picture of the expectations and knowledge of the users and to adapt the interfaces to this knowledge [GWGT10]. The information presented to the user should not be complex, but should contain clearly structured information. In order to reduce complexity, only important information should be presented using self-explanatory visual concepts such as icons [HHN10]. Any software can present great approaches and ideas, but as long as users do not understand it, they will not accept and use it. Therefore strong visual concepts should be employed; supportive visualisation techniques such as icons present PETs in a more understandable way. This knowledge is also supported by our findings in the PrimeLife project (e.g., research on privacy icons [Pri09c]).

## **11.4 HCI Activities and Software Development**

In Section 11.3, we described how we have addressed the challenges within the PrimeLife project on an abstract level. Given the nature of the different prototypes and the different HCI approaches, not all challenges could be addressed to the same extent for each prototype. In practice, there are different ways of including HCI work in software development processes. In general, HCI engineers have to adopt their methods and fine-tune them to fit different development styles such as extreme programming [WTS<sup>+</sup>08].

This section focuses on the process perspective and shows the practical integration of the above introduced challenges into the PET software development processes in PrimeLife.

### ***11.4.1 Backup Prototype***

The main purpose of the backup prototype is to give the user the possibility to create backups of data and delegate access rights. For example, in case of illness the user is able to delegate access rights to other persons, which is an important privacy issue.

The development of the backup prototype as part of the PrimeLife project can be considered as a best case approach to solve the challenges introduced above. It was user-centered and HCI driven from the very beginning. In fact, the entire specification of the prototype was based on user research (e.g., requirements) and completed by HCI engineers.

For example, Challenge 1: limited end user knowledge of PETs was addressed through the development of different views in the interface (standard view and expert view) based on gathered user requirements. The standard view provides all necessary elements for creating backups and delegating access rights. The expert view should provide in-depth settings for backups and delegation, mainly used by expert users.

When the user interface design of this prototype started, no technical specifications were available. Therefore the whole UI was driven by a strong focus on user needs. Furthermore, we focused on human-computer-interaction paradigms and usability guidelines; possible technical matters were ignored during the first part of the UI design. In the second iteration, we adapted the developed prototype to the technical specifications without changing the interaction paradigms. Thus, the entire UI assists the user in understanding how the system works and does not need any special knowledge concerning PETs and privacy from the user, answering directly to Challenge 2: technologically driven development of PETs. Additionally, we were focussing on the usage of understandable terms, mainly employing common terms and explaining novice and unknown terms (Challenge 3: understanding of PET related terms).

To reduce the cognitive load of users when dealing with this application, we concentrated on the usage of elements, in addition to the terms, which are familiar to users. We also made the workflow of the tool transparent for users by showing them consequences of providing unnecessary or technical informations. Therefore we hoped to correct wrong mental models by answering to Challenge 4: wrong mental models of PETs.

As mentioned before, the user interface of the backup prototype is based on the needs of the users and not on technical requirements. The design of the prototype presents the workflow and interaction possibilities as easy as possible although the functionality behind the algorithms is very complex (Challenge 6: Complex mechanisms are hard to understand).

Thus, user involvement was assured and several of the above introduced challenges, such as problems in understanding privacy and PET related terms as well as mental models and complex structures could be addressed and counteracted with the above introduced methods at a very early stage of development.

### ***11.4.2 Privacy Dashboard***

The privacy dashboard indicates the security of websites and allows for fine-grained handling of privacy information. In this approach, the HCI engineer did not design



mock-ups but supported implementation by directly working on the source code of the user interface for the privacy dashboard<sup>1</sup>.

In the development of the privacy dashboard, the limited user knowledge of PETs (Challenge 1) was counteracted by providing, in addition to the standard view, personalisation possibilities for each website and by introducing awareness mechanisms to the user. To ease understanding of PET related terms (Challenge 3), the use of PET particular wording as well as abbreviations was avoided (de-technification of terms) and definitions and explanations were provided for PET-related terminology (e.g., “third party cookie”). Additionally, the results of the privacy terms study [GWKT10] were used to distinguish easily understandable terms from PET-related wording.

Challenge 2: technologically driven development of PETs was met through permanent heuristic evaluations of the developed prototypes and user evaluations. Thus, direct feedback either from experts or from users could be included in the development process, permanently advancing the quality of the prototype.

For the privacy dashboard, it was particularly important to create an interface that is easy to handle and does not require much time to do so, answering Challenge 5: privacy as a secondary task. Therefore, the main interaction when visiting a new website was reduced to two clicks, where the user can adjust his privacy settings. The settings will then be saved for further visits to the website. Furthermore, the risks of a website are presented in an understandable way (even if complex mechanisms are described). Therefore Challenge 6: complex mechanisms are hard to understand was also addressed by the privacy dashboard.

The design approach described here was experienced to be very effective. The precondition to enable an efficient development workflow also applied to the privacy dashboard is that relevant tools and workflows are decided upon and put in place (e.g., SVN for managing the source files to ensure that re-engineered UI-code will be part of the iterations). This also requires the HCI engineer to have knowledge of the used programming and markup languages, which is not always the case.

### ***11.4.3 Examples Reflected***

In the above sections, we showed different approaches on how to answer the challenges described in Section 11.2 during the development of two PrimeLife prototypes. In practice, the meeting of challenges depends on a number of criteria, such as the collaboration between developers and HCI engineers or the fidelity of the prototype when HCI activities are started.

The experience of the PrimeLife project showed that early inclusion of HCI knowledge is a key success factor that provides answers to many of the above introduced challenges. For PETs themselves, it is not utterly important which UCD process is applied, as long as HCI activities are adopted as early as possible. The

---

<sup>1</sup> <http://www.primelife.eu/results/opensource/76-dashboard>

kind of HCI evaluations conducted and the feedback provided to development are important.

## 11.5 Discussion and Outlook

In this chapter, we demonstrated that current PETs lack usability and when developing PETs one will encounter several of the above-mentioned challenges, since the entire field of PETs is currently still in an early stage of development. This goes hand in hand with knowledge deficits of users: after multiple iterations of user evaluations, we can conclude that the lack of privacy knowledge calls for a focus on mental models and users' understanding of terms and background mechanics, which is the focus of our adoptions to the UCD process. Based on our experience during the PrimeLife project, we presented several approaches and practical examples on how common UCD processes can be enhanced and adapted to PETs.

As the field of ICT and PETs in particular is developing very fast, we expect the suggested processes to be used and challenges to be adopted quickly within the near future. This will also foster the adoption of PETs by a wider public and hence lead to "privacy by design."

As an outlook on possible future methods for evaluations of PETs, we provide a review of user involvement methods defined in ISO/TR 16982:2002(E) "Ergonomics of human-system interaction—Usability methods supporting human-centered design:"

- **Observation of users:** Systematic collection of observation material is done in usability laboratory evaluations. Test leaders are used to observe not only what users are doing but also how they react emotionally. For PET evaluation, we recommend having an experienced test leader observe. This is because non-verbal cues are much more important in discovering a user's attitudes towards PETs than towards default software products such as websites.
- **Performance related measurements:** It is important to implement special PET performance measures. The "classical" performance measures (task completion time, error rate) should be extended with data disclosure measures. In combination with questionnaires and/or interviews, such measurements will provide the needed knowledge on how to communicate complex privacy issues.
- **Critical incidents analysis:** Critical incidents in the PET domain, as we observed during the PrimeLife project, are a welcome extension to the error rate measurement.
- **Questionnaires/interviews:** When answering questions in the HCI/PET domain, users will not be able to express how they will behave when using the software. Questionnaires/interviews are worthy when measurements and/or observations provide data to base the analysis on. For example, the users can be asked with a questionnaire or during an interview which private data they think they have disclosed to others. The technical measurement (a log file analysis in this case) will tell if the PET succeeded in supporting the user to preserve privacy or not.

- **Thinking aloud:** This method, where users continuously verbalise their ideas, beliefs, expectations, doubts, discoveries, etc. during the evaluation, is well-suited to give insight into the mental models of the users. We recommend including the thinking aloud methodology as a default to every PET design.
- **Collaborative design and evaluation/creativity methods:** Due to the fact that knowledge about PETs is neither much distributed on a wide scale nor broadly available, collaborative design and creativity methods are not very helpful from our experience. But we assume that collaborative design with a focus on the mediation between user and developer seems to be a promising approach. Therefore, further research is needed on this issue.

## Chapter 12

# The Users' Mental Models' Effect on their Comprehension of Anonymous Credentials

Erik Wästlund and Simone Fischer-Hübner

**Abstract** Anonymous Credentials are a key technology for enforcing data minimisation for online applications. The design of easily understandable user interfaces for the use of anonymous credentials is however a major challenge, as end users are not yet familiar with this rather new and complex technology and no obvious real-world analogies exist for them. In this chapter, we analyse what effects the users' mental models have on their understanding of the data minimization property of anonymous credentials in the context of an e-Shopping application scenario. In particular, we have investigated the effects of the mental models of a card-based user interface approach and an attribute-based user interface approach and compared these in terms of errors of omission and addition. The results show that the card-based approach leads to significantly more errors of addition (i.e., users believe that they have disclosed more information than they actually have) whereas the attribute-based approach leads to more errors of omission (i.e., users underestimate the amount of data that they have disclosed).

## 12.1 Introduction

A fundamental privacy design principle is data minimisation, meaning that services or applications should be designed in accordance with the aim of collecting and processing as little personal data as possible. Data minimisation limits the communication partner's ability to profile users and is as a legal principle well acknowledged by most Western privacy laws. It can in particular be derived from Art. 6 I (c), 6 I (e) of the EU Data Protection Directive 95/46/EC [Dir95] and is for instance also required explicitly by Section 3a of the German Federal Data protection Act [Ger09]. Anonymous credentials are a key technology for achieving data minimisation on an application level. The aim of this chapter is to show our work with creating UIs (user interfaces) with the objective to make users comprehend the data minimization property of anonymous credentials. Anonymous credentials

are becoming increasingly significant not only in research but also in practice: The identity mixer (Idemix) anonymous credential protocol [CL01] used in PrimeLife is contributed by IBM Open Source and Microsoft has been incorporating the U-Prove anonymous credential technology into Windows Communication Foundation and Windows CardSpace. Mental models can be understood as thought models of how a system works. The mental model of a system is often divided into three parts; the view of the programmers, the view of the user, and the UI design [Coo95] [Nor88]. In this chapter, we will describe the effects of inducing different mental models on UIs that support data minimisation as well as various comprehension problems that users have due to their mental models of the card-based metaphor and of privacy-enhanced e-Shopping transactions in general.

### ***12.1.1 Anonymous Credentials***

A traditional credential (often also called certificate or attribute certificate) is a set of personal attributes, such as date of birth, name or personal number, signed (and thereby certified) by the certifying party and bound to its owner by cryptographic means (e.g., by requiring the owner's secret key to use the credential). In terms of privacy, the use of (traditional or anonymous) credentials is better than a direct request to a certifying party, as this prevents the certifying party from profiling the user [CSS<sup>+</sup>05]. Traditional credentials require, however, that all attributes are disclosed together if the owner wants to prove certain properties. This makes different uses of the same credential linkable to each other. Anonymous credentials (also called private certificates) were first introduced by Chaum [Cha85] and later enhanced by Brands [Bra99] and by Camenisch and Lysyanskaya and their Idemix protocol [CL01] and have stronger privacy properties than traditional credentials. Anonymous credentials, in contrast to traditional credentials, allow a user to selectively reveal only a subset of her attributes or to prove that she has a credential with specific properties without revealing the credential itself or any additional information. For example, if a user has a governmentally issued anonymous passport credential with personal attributes typically stored in a passport including her date of birth and she wants to purchase a video online which is only permitted for adults, she can prove with her credential via a cryptographic zero-knowledge proof just the fact that she is older than 18 without revealing her date of birth or any other attributes of her credential. In another scenario, the holder can use her anonymous passport credential for proving her gender or name. In other words, anonymous credentials allow the selective disclosure of identity information encoded into the credential. In addition, the Idemix anonymous credential system also has the property that multiple uses of the same credential cannot be linked to each other, which can prevent the linking and profiling of different user sessions. If, for instance, the user later wants to buy another video which is only permitted for adults at the same online video store, she can use the same anonymous credential as proof that she is over 18 without the video store being able to recognise that the two proofs are based on the same

credential. This means that the two rental transactions cannot be linked to the same person. Hence, whereas traditional electronic credentials have very similar properties as real-world credentials, anonymous have further data minimisation characteristics that differ from the ones of real-world credentials. A special challenge for HCI (Human Computer Interaction) representations of anonymous credentials is to illustrate the following two characteristics:

- *Selective Disclosure*: Proving only some of the attributes of an anonymous credential.
- *Unlinkability*: Multiple uses of the same anonymous credentials cannot be linked

The main focus of our empirical Usability studies, which we present in this paper, has so far been on the comprehension of the selective data disclosure property.

### 12.1.2 Related Work

Within the scope of the PRIME<sup>1</sup> project, our usability tests of PRIME prototypes revealed that users often did not trust privacy-enhancing technologies and their data minimisation properties, as the possibility to use Internet services anonymously did not fit to their mental model of Internet technology [PFHD<sup>+</sup>05] [ACC<sup>+</sup>ce]. Camenisch et al. [CSSZ06] discuss contextual, browser-integrated user interfaces for using anonymous credential systems. In user tests of anonymous credential selection mockups developed within the PRIME project, test subjects were asked to explain what information was actually given to a web site that demanded some proof of age when a *passport* was used to produce that proof (more precisely, the phrase “*Proof of “age > 18” [built on “Swedish Passport”]*” was used as a menu selection choice in the mockup). The test results showed that the test users assumed that all data normally visible in the physical item referred to (i.e., a passport) was also disclosed to the web site [WP008]. Hence, previous HCI studies in the PRIME project showed already that designing user interfaces supporting the comprehension of anonymous credentials and their selective disclosure property is a challenging task.

More than 10 years ago, Whitten and Tygar [WT99] discussed the related problem that the standard model of user interface design is not sufficient to make computer security usable to people who are not already knowledgeable in that area. They conclude that a valid conceptual model of security has to be established and must be quickly and effectively communicated to the user.

---

<sup>1</sup> EU FP6 integrated project PRIME (Privacy and Identity Management for Europe), <https://www.prime-project.eu/>

## 12.2 Performed User Tests

The issue of anonymous credential selection paradigms and mental models is tightly knit to users' current understanding and usage of ordinary plastic credit- and identity cards. On good grounds, proponents of the plastic card paradigm argue that this is easily understood by users as it mimics what they already know and use every day. Thus, explaining the basic idea of digital credentials is easily done by comparing them to their plastic predecessors. The drawback of this metaphor is that plastic cards do not lend themselves to selective disclosure. An alternative to the card-based metaphor is the attribute-based approach where no identity attributes are referenced as specific entities other than belonging to a specific group i.e., card. In order to investigate which of the two paradigms makes it easier for users to understand the principle of data-minimisation, two lines of variations of UIs were proposed. The UIs differed in terms of selection mechanism. The selection was either done by clicking on a representation of the full card containing the desired attribute or by choosing the attribute from a dropdown list containing the possible sources of the attribute.

### 12.2.1 Method

All the user tests presented in this study have been conducted in the same fashion. The users have been introduced to the principle of selective disclosure, the system they are about to work with, and the eShopping task they are about to perform. After having performed their tasks, users were asked to describe what personal information they had conveyed to the data recipient. Our main interest with that question was to analyse if they understood the principle of selective disclosure or if the users would believe that they had sent additional information.

#### 12.2.1.1 Participants and Test Design

The participants of the user tests were students enrolled at Karlstad University (with the exception of a few employees), aged between 18 and 65, with the vast majority being under 35. The participants were volunteers recruited around the campus and as participation was not part of any specific course, the participants studied a variety of subjects. The tested UIs have been developed through a cyclic process of UI prototyping, usability testing, and subsequent refinements and redesigns of the user interfaces. Most of the user tests are small sample studies and include only five participants. Nielsen [Nie00] points out that while experiments showed that a test with at least 15 users is needed to discover all the usability problems in the design, it is better to distribute the budget for user testing across several iterations of mockup developments/improvements and tests, e.g., it will be better to spend this budget on three tests with 5 users each. After the first study with 5 users, usually 85% of

the usability problems will be found, and it is recommended to fix these problems in a redesign before testing again. For this reason, we have also decided to do a series of iterations of mockups redesigns for our policy mockups and tested the early iterations with only 5 test persons in each iteration round. Having said this, it is important to point out that the statistical analysis is done on a combined level, contrasting all UI's in the respective approach with 35 users testing card-based UIs and 48 users testing the attribute-based UIs.

Independent of the credential selection paradigm, the general setting of the tests was identical. In order to create a realistic experience for the test participants, we created an interactive Firefox background for the user tests. The background consisted of an image of the Firefox web browser, which in turn contained a scrollable image of the buy kindle books section of the Amazon.com web page. The Amazon.com image was equipped with clickable links that would activate the pop up of the select credentials console and, thus, test participants would get the familiar look and feel of the web site as it looks today up to the point of paying for the purchase. After the test participants had been introduced to the basic principles of selective disclosure, they were asked to purchase a book of their own choice with the credentials that were available to them at the time. After the purchase they were asked what data they had given to Amazon.com during the transaction.

The users were all given the same scenario. They were to act as Inga Vainstein who had the anonymous credential system installed on her machine and use her credentials to buy a book from Amazon.com. Inga had already imported anonymous credentials from the Swedish passport authority and the Swedish road authority and had also entered information about her Visa and Amex credit cards. The participants were told that they were to test a new and more secure payment system where they had to prove their right to use a credit card by proving that they had the same name as the credit card holder by using one of the two anonymous credentials. Independent of the paradigm, the correct response was that the only information they had conveyed to the data recipient was their name and the issuer. However, from the information about the issuer, further meta information can be derived, e.g., Inga is a Swedish citizen or Inga has a Swedish driver license. Not reporting the issuer (or any other information that was sent in the scenario) has been recorded as an error of omission, while reporting that more has been revealed has been recorded as an error of addition. Apart from the differences in UIs, there was also a difference in how the users gave their responses. In the card-based scenarios, the users were shown images of the source cards together with a table containing all the information on the cards and they were instructed to tick the information being conveyed. As no reference was made to cards in the attribute-based scenario, instead of showing the source cards, the participants simply got to write their response on a piece of paper.



## 12.2.2 The Card-Based Approach

In order to investigate how an effective selection mechanism for a card-based identity management system should be designed, we tested a number of different alternatives in different iterations. The reason for the many iterations<sup>2</sup> was the simple fact that very few of the participants understood the principle of selective disclosure.

The selection mechanism of our first credential selection UI showed the full cards and once the participants had chosen both a credit card and identity card it was possible to continue to the summary page which contained only the selected information. As none of the participants understood the concept of data minimisation, we explored a number of alternative UIs, all with the central idea to show what information was selected from the different source cards. Additionally, during some of the tests rounds, we instructed the users to create a new temporary (virtual) card instead of selecting cards in order to move them out of their current understating of credentials.

Our first alternative UI was based on the idea of highlighting and explicitly showing that some information is selected and that some is not. In order to do so, we added a mouse over state where the information about to be selected was shown with little scissors around cut outs and the background was blurred out (See [Figure 12.1](#) for an example).

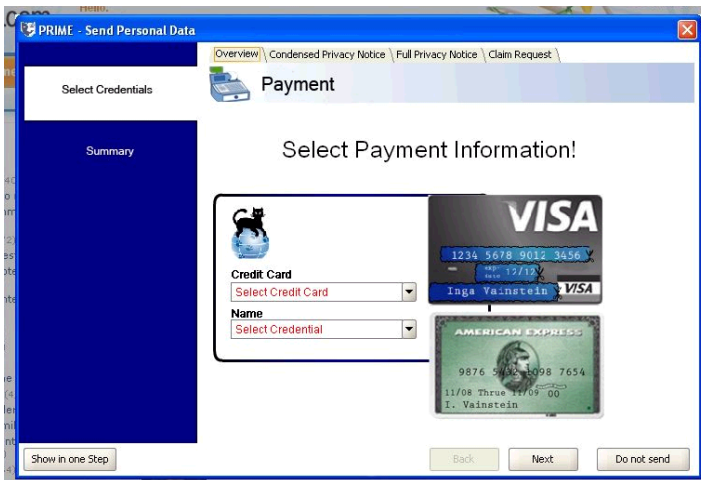


Fig. 12.1: The Cut out and blur UI with the Visa card in its mouse over state.

In the next iteration, we created a short animation where the cutout, when clicked, moved from the source card to the virtual card while the source card dissolved. This was done based on the idea that the error was due to users perceiving the grayed out

<sup>2</sup> For a description of the reasoning behind the iterations see [Pri09a, Chapter 5]

information as being sent but not being important. The animation however, did not help the users to understand the selective disclosure as our tests revealed.

Despite our efforts to highlight that specific data was being selected from the source cards, users still overestimated the amount of data being sent. One possible explanation was that users, having seen the data, associated it with the cards and thus thought it was being sent, we decided to completely hide all data that were not sent and to show only relevant data that was about to be selected (See [Figure 12.2](#) for an example).

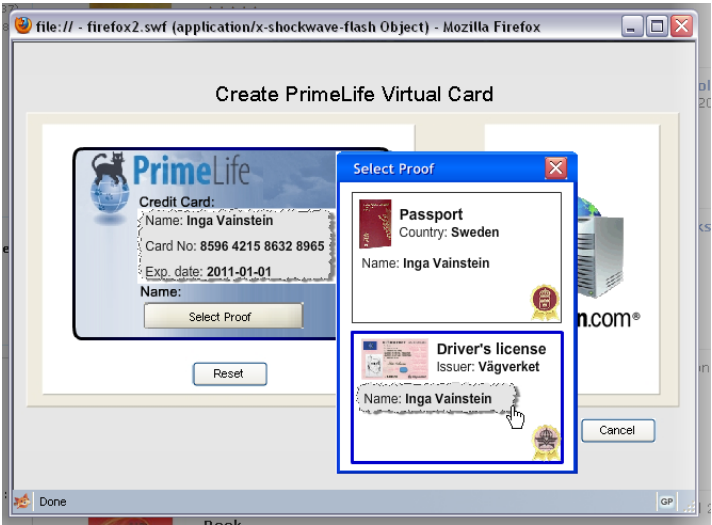


Fig. 12.2: UI that shows only relevant data.

Even though the full cards were not shown, the majority of users thought that all of the data they associated with the source cards were sent. Therefore we focused on making it very explicit that the data from the source cards was suppressed by utilising the familiar concept of blacking out information (See [Figure 12.3](#) for an example).

Still, most users believed they had sent all data from the source card. One possible reason for this might be that users believe that the blacked out data is being sent in encrypted form. A more practical problem with this design is the fact that credentials that contain a lot of information (e.g., passports) take up a lot of screen space, which makes it difficult to show if the transaction at hand demands information from multiple sources.



Fig. 12.3: The summary page of UI blacking out data that are not sent.

12.2.2.1 Test Results for the Card-Based Selection Paradigm

All in all, we performed seven iteration rounds of tests with subsequent UI improvements, where 5 users participated in each test round. Of the 35 users, only two understood that the only information they sent from the driver’s license or passport was the name (Inga Vainstein) plus information about the card issuer (in form of the signature by the Swedish government or Swedish road authority). Further, three test users understood the principle of selective disclosure in so far as they understood that the name and not all information was sent. However, they missed the fact that information about the card issuer was revealed as well. Thus, five users (14%) understood, at least up to a point, the principle of selective disclosure with the card-based approach.

12.2.3 The Attribute-Based Approach

As the majority of errors during testing of the card-based UIs were due to users not understanding the principle of selective disclosure, we wanted to know if users would perform better or if errors are different if there is no mention of cards in the UIs. Instead of informing the users that they had imported credentials in the form of

cards, we told the users that they had imported validated attributes of information from the Swedish passport authority and the Swedish road authority.

The first iteration of UIs contained two drop down lists where the user could select credit card and verifier of proof. Once the user had selected one of each, it became possible to click the “send” button. (See [Figure 12.4](#) for an example).



Fig. 12.4: UI for selecting information and verifier.

A common user comment regarded the labeling Proof [Name: Inga Vainstein] and verifier. This was addressed in the following iteration where the user was instead asked to select issuer of the proof. Although these changes led to a substantially higher amount of correct responses in terms of selective disclosure, the users instead did not understand that the information about the issuers was also being sent.

### 12.2.3.1 Test Results for the Attribute-based Selection Paradigm

All in all, we performed six iterations of tests with an average of 8.5 users in each. Of the 48 users, 22 understood the selective disclosure and correctly stated that name and issuer was sent. Another 10 users missed the disclosure of the issuer but understood that nothing more than their name was sent. In total, 32 users (66%) fully or partially understood the principle of selective disclosure. Another interesting observation is that amongst the users who used the first attribute-based UI with the 'select verifier' instruction, some interpreted the instruction as the data being sent via the verifier who would then be able to trace all transactions being made by the user. Hence, they got the wrong impression that the verifier (e.g., the police or the Swedish road authorities) could in the end trace their activities.

12.2.4 Results of the User Studies

In order to answer the main question of this study, namely, does the user’s mental model affect the understanding of selective data disclosure, a binominal test was performed between the results of the card-based approach and the attribute-based approach. The results of the binominal test showed that there is a significant difference ( $p < 0.001$ ) between the 14% correct response rate in the card-based approach ( $n = 35$ ) and the correct response rate, 66% in the attribute-based approach ( $n = 48$ ). It should be noted that in this analysis both users omitting issuers have been counted as correct (for specific proportions see [Table 12.1](#)).

Table 12.1: Proportions (and count) of errors of omission, correct responses and errors of addition in the card-based and the attribute-based approach.

	Omission	Correct	Addition
Card	9% (3)	6% (2)	86% (30)
Attribute	21% (10)	46% (22)	33% (16)

A noteworthy fact is the difference between the two approaches in terms of errors. Whereas there were ten times as many errors of addition than errors of omission in the card-based approach, the same factor for the attribute-based approach was 1.6. The main source of errors of addition in the card-based approach was the notion that all information of the source card is sent. A few users also suggested that now Amazon even knows what they look like as well as their handwriting, indicating that these users believed an exact copy of the source card was sent. A contrasting mental model artifact was observed in the attribute-based approach. However, instead of the concept of the card, it was the concept of the verifier that led users down the wrong path believing that data was sent via the verifier who would then gain knowledge of the transaction. Moreover, in the attribute-based approach, the main source of errors of addition was personal identification number and address, even though these attributes were not part of the eShopping scenario. As the use of personal identification numbers is however very common in Sweden, this error is most probably a cultural artifact.<sup>3</sup> Being a “very Swedish” error or not, it still shows that the users had the wrong mental model in regards to privacy-enhance e-Shopping with anonymous credentials.

12.3 Conclusions & Future Work

The main result of this study is that the mental models of users affect their understanding of the selective disclosure property of anonymous credentials. Further-

<sup>3</sup> In Sweden, personal identification numbers are extensively used both in contacts with government agencies and private companies in all situations where an individual is to be identified.

more, we have shown that the use of an attribute-based approach to the design of a credential selection interface leads to fewer misunderstandings in terms of data disclosure than when the UI design uses the card-based approach. It should however be stated that even though users made less errors of addition in the attribute-based approach, we do not know if this is due to the fact that they actually understood the principle of selective disclosure or if they simply looked at the UI and made their inferences. The results of the attribute based user tests indicate that it is in fact the latter as there were a number of users not understanding that the issuer was revealed or that believed that their personal identification number was revealed as well. A relevant question is of course what the objective of the UI is: should it educate the user of the properties of anonymous credentials or is it enough that the users understand the end result of using such a system?

In this study we did not include any aspects of unlinkability. However, intuitively, the attribute-based approach seems to be more appropriate for meditating the unlinkability property of anonymous credentials than the card-based approach, as in the real world the different displays of physical cards are linkable. Further research needs to be conducted on this question.

As the key issue for the deployment of privacy enhancing technology such as the use of anonymous credentials is to have users understand the privacy-enhancing features, or at least not misunderstand them, the mental model invoked by the UI is of great importance. Hence, the card-based approach can only be successfully deployed, if we can fill the gap between the card-in-your-wallet approach and the functionality of anonymous credentials.

One possible way to solve this problem is to focus on the main difference between the cards and the anonymous credentials, namely that the latter can be adapted to fit user's current needs. Future research being planned within the last year of the PrimeLife project will investigate the effects of introducing the credential selection mechanism as being part of an adaptable eID system on the user's understanding of data minimisation. This research will also include questions regarding the unlinkability of credentials. Last but not least, in addition to investigating the users' mental models with regards to their understanding of data minimisation, we will also investigate users' understanding of how their personal data flows between the credential verifier and the services side. Taken together, users can only value the privacy features of anonymous credentials completely, if they can understand their minimal disclosure and unlinkability properties, and if they can understand that personal data can flow directly to a services side without the involvement of the credential issuer.

## 12.4 Acknowledgments

We would like to thank our colleagues within the PrimeLife consortium for their cooperation. Jenny Nilsson, Maria Lindström and Erica Resare from KAU, who conducted some of the tests. John Sören Pettersson, Hans Hedbom (KAU), Peter Wolkerstorfer, Christina Köffel (CURE), Franz-Stefan Preiss, Patrik Bichsel, Gre-

gory Neven, Jan Camenisch (IBM Research – Zurich) as well as Harald Zwingelberg and Marit Hansen (ULD) who all participated in discussions of our mockups and provided helpful feedback.

# Chapter 13

## Trust and Assurance HCI

Simone Fischer-Hübner, Hans Hedbom, and Erik Wästlund

**Abstract** In this chapter, we present our HCI (Human Computer Interaction) work for mediating the degree of trustworthiness of services sides to end users and for enhancing their trust in PrimeLife-enabled applications. For this, we will present the user interface development work of a trust evaluation function and the PrimeLife Data Track.

### 13.1 Introduction

Trust plays an important role for PrimeLife, because users do not only need to trust their own platforms and user side identity management components to manage their data properly. They also need to trust communication partners and their remote set of platforms that receive personal data to process their data in a privacy-friendly manner and according to (business) agreements. Our previous HCI work within the FP6 project PRIME, as well as the research by others, had revealed that end users often do not trust the claims of the privacy-enhancing features of privacy-enhancing technologies (PETs)[PFHD<sup>+</sup>05], [SGB01]. Within PrimeLife, we have therefore conducted research on how user interfaces can contribute to communicate the trustworthiness of PrimeLife technologies and assurance information to the end users.

In this context, we have first investigated social trust factors and, based on this, developed and tested user interface (UI) mockups for a trust evaluation function, which uses a multi-layer design for informing users about the evaluation of a services side's trustworthiness in regard to privacy and business reliability.

Additionally, we have developed the Data Track, which provides the user with a history function documenting what personal data the user has revealed under which conditions, and includes online functions for a user to exercise her rights to access/-correct/delete her data at the remote data controller's side (cf. Art. 10 EU Directive 95/46/EC). As research on social trust factors has shown, trust in online transactions



can be increased if the transactions are transparent and reversible (see below). The Data Track is a transparency-enhancing tool, which can thus increase the end users' trust and assurance that their personal data are handled properly by others and can, with its online functions, enhance the user's control over her personal spheres.

In the chapter, we will first in Section 13.2 discuss social trust factors, which have motivated our design of the trust evaluation function and the Data Track. Then, we will in section 13.3 present the design and tests of the trust evaluation function. Results of these first two sections have also been presented earlier in [FHFL09]. In Section 13.4, we will discuss the user interface designs for Data Track that we have developed for the PrimeLife project as well as the usability test results from tests that we conducted at Karlstad University and at the Centre for Usability Research in Austria. Finally, conclusions are drawn at the end of this chapter.

## 13.2 Social Trust Factors

In this section, we investigate suitable parameters corresponding to social trust factors for measuring the actual trustworthiness of a communication partner in terms of privacy practices and of the reliability as a business partner and for establishing reliable trust.

Social trust factors in the context of e-Commerce have already been researched by others. For instance, [TZY01] showed that for ordinary users to feel secure when transacting with a website, the following factors play a role: 1. the company's reputation, 2. their experiences with the website, and 3. recommendations from independent third parties.

Riegelsberger et al. [RSM05] present a trust framework that is based on contextual properties (based on temporal, social and institutional embeddedness) and the services side's intrinsic properties (ability, motivation based on internalised norms, such as privacy policies, and benevolence) that form the basis of trustworthy behaviour. Temporal embeddedness can be signalled by visible investments in the business and the side, as e.g. visualised by professional website design, which can also be seen a symptom for the vendor's intrinsic property of competence or ability to fulfill a contract. Social embeddedness, i.e. the exchange of information about a side's performance among users, can be addressed by reputation systems. Institutional embeddedness refers to the assurance of trustworthiness by institutions, as done with trust seal programs.

A model of social trust factors, which was developed by social science researchers in the PRIME project [LLPH05], [ACC<sup>+</sup>ce], has identified 5 layers on which trust plays a role in online services: socio-cultural, institutional, service area, application, and media. Service area-related trust aspects that concern the trust put in a particular branch or sector of economic activity, as well as socio-cultural trust aspects, cannot however be directly influenced by system designers. More suitable factors for establishing reliable trust can be achieved on the institutional and application layers of the model, which also refer to trust properties (contextual property

based on the institutional embeddedness as well as certain intrinsic properties of a web application) of the framework by Riegelsberger et al. [RSM05]. As discussed by Leenes et al. [ACC<sup>+</sup>ce], on the institutional layer, trust in a service provider can be established by monitoring and enforcing institutions, such as data protection commissioners, consumer organisations and certification bodies. Also, on the application layer, trust in an application can be enhanced if procedures are clear, transparent and reversible, so that users feel in control.

This latter finding also corresponds to the results of the British Trustguide project [LCP06], which also provides guidelines on how cybertrust can be enhanced and also concludes that increased transparency brings increased user confidence.

### 13.3 A Trust Evaluation Function

In this section, we will present a trust evaluation function that has been developed within the PrimeLife EU project (see also [Pri09b], [FHFL09]). This function has the purpose of communicating reliable information about trustworthiness and assurance (that the stated privacy functionality is provided) of services sides. For the design of this trust evaluation function, we have followed an interdisciplinary approach by investigating social factors for establishing reliable trust, technical and organisational means, as well as HCI concepts for mediating evaluation results to the end users.

#### 13.3.1 *Trust Parameters Used*

Taking results of the studies on social trust factors presented in Section 13.3 as well as available technical and organisational means into consideration, we have chosen the following parameters for evaluating the trustworthiness of communication partners that mainly refer to the institutional and application layers of the social trust factor model. Information provided by trustworthy independent monitoring and enforcing institutions, which we are utilising for our trust evaluation function, comprise:

- Privacy and trust seals certified by data protection commissioners or independent certifiers (e.g., the EuroPrise seal, the TRUSTe seal or the ULD Gütesiegel);
- Blacklists maintained by consumer organisations (such blacklists exist for example in Sweden and Denmark);
- Privacy and security alert lists, such as list of alerts raised by data protection commissioners or Google's anti-phishing blacklist.

It is important to note that black lists and alert lists have to be carefully chosen to ensure that blacklisting and alert listings are based on fair decisions. For this, it has to be checked how reliable the list operators are, who is controlling them and what

are the criteria for blacklisting or alert listing. The European Consumer Centres have launched a web-based solution, *Howard the owl*, for checking trust marks and other signs of trustworthiness that could be used as well when evaluating a web shop .

Static seals can be complemented by dynamic (in real-time generated) seals conveying assurance information about the current security state of the services side's system and its implemented privacy and security functions. Such dynamic seals can be generated in real-time by an "Assurance Evaluation" component that has been implemented within the PRIME framework [Pea06]. Dynamic seals that are generated by tamper-resistant hardware can be regarded as third-party endorsed assurances, as the tamper-resistant hardware device can be modeled as a third party that is not under full control of the services side. Such dynamic assurance seals can measure the intrinsic property of a side's benevolence to implement privacy-enhancing functionality. Such privacy-enhancing functionality can also comprise transparency-enhancing tools that allow users to access, and to request to rectify or delete their personal data online (cf. Data Track described in Section 13.4), which will allow users to "undo" personal data releases and to feel in control. As discussed above, this is an important prerequisite for establishing trust. For our trust evaluation function, we therefore used dynamic assurance seals informing about the PrimeLife privacy-enhancing functions that the services side's system has implemented. Also, reputation metrics based on other users' ratings can influence user trust, as discussed above. Reputation systems, such as for instance the one on eBay, can, however, often be manipulated by reputation forging or poisoning. Besides, the calculated reputation values are often based on subjective ratings by non-experts, for whom it might be difficult to judge the privacy-friendliness of communication partners. We have therefore not considered reputation metrics for the PrimeLife trust evaluation function.

Following the process of trust and policy negotiation of the PRIME technical architectures (on which also PrimeLife systems are based), privacy seals, which are digitally signed by the issuing institution, as well as dynamic assurance seals can be requested from a services side directly (see steps 4-5 in [Figure 13.1](#)), whereas information about blacklisting and alerts need to be retrieved from the third party list providers (see steps 6-7 in [Figure 13.1](#)). After the user requests a service (step 1), the services side replies with a request of personal data and a proposal of a privacy policy (step 2). For evaluating the side's trustworthiness, the user can then in turn request trust and assurance data and evidences from the services side, such as privacy seals and dynamic assurance seals (Steps 4-5), and information about blacklisting or alerts concerning this side from alert list or blacklist providers (Steps 6-7). Information about the requested trust parameters are then evaluated at the user side and displayed via the trust evaluation user interfaces along with the privacy policy information of the services side within the "Send Personal Data?" dialogue window (see below), with which also the user's informed consent for releasing the requested data for the stated policy is solicited. The user can then, based on the trust evaluation results and policy information, decide on releasing the requested personal data items and possibly adopt the proposed policy, which is then sent to the service provider (Step 8).

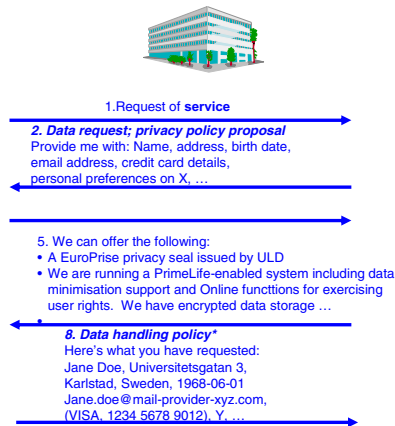


Fig. 13.1: The steps for Privacy and Trust negotiation as proposed by PrimeLife.

### 13.3.2 Design Principles and Test Results

For the design of our trust evaluation function mock-ups, we followed the following design principles comprising general HCI principles as well as design principles, which should in particular address challenges and usability problems that we have encountered in previous usability tests:

- Use a Multi-layered structure for displaying evaluation results, i.e. trust evaluation results should be displayed in increasing details on multiple layers in order to prevent an information overload for users not interested in the details or the evaluation. Our mockups have been structured into three layers, displaying a short status view with the overall evaluations for inclusion in status bars and in the “Send Personal Data?” window (1st layer, see [Figure 13.2](#)) displaying also the services side’s short privacy policy and data request; a compressed view displaying the overall results within the categories privacy seals, privacy & security alert lists, support of PRIME functions and blacklisting (2nd layer); and a complete view showing the results of sub categories (3rd layer, see [Figure 13.3](#)).
- Use a selection of meaningful overall evaluation results. For example, in our mockups, we use a trust meter with a range of three possible overall evaluation results that provide a semantic by their names (which should be more meaningful

The screenshot shows a web window titled "Send Personal Data ?". Inside, there are three main sections:

- Your data:** A box containing the text "Inga Vainstein, Kungsgatan 12, Karlstad VISA 567 899 000 222-987".
- ...is requested by:** A box containing "Movies Inc" with the "We Love Movies!" logo, the URL "www.welovemovies.com", and a "Trust Evaluation result" meter. The meter is a semi-circle with a red section (Poor), a white section (Fair), and a green section (Good). The needle points to the "Fair" section.
- Purposes:** A box containing the text "Betaltning och leverans av beställd film". To the right of this box is a link: "Link to [full privacy policy](#)".

At the bottom right, there are two buttons: "Send" and "Cancel".

Fig. 13.2: “Send Personal Data?” window displaying the overall trust evaluation result (1st Layer).

than for instance percentages as used by some reputation metrics). The three overall results that we are using are (see trust meter in [Figure 13.2](#)):

- “Poor” symbolised with a sad-looking emoticon and red background colour (if there are negative evaluation results, i.e. the side is blacklisted or appears on alert lists);
  - “Good” symbolised with a happy looking smiley and green background colour (if there are no negative, but some positive results, i.e. the side has a seal or supports PrimeLife functions and is not appearing on black/alert lists);
  - “Fair” symbolised with a white background colour (for all other cases, i.e. the side has no seal, does not support PrimeLife functions, and does not appear on black/alert lists).
- Make clear who is evaluated - this is especially important, because as we mentioned above, our previous usability tests have revealed that users often have difficulties to differentiate between user and services side [PFHD<sup>+</sup>05]. Hence, the user interface should make clear by its structure (e.g., by surrounding all information referring to a requesting services side, as illustrated in the “Send Personal

Data?” window [Figure 13.2](#)), and by wording that the services side and not the user side is evaluated. If this is not made clear, a bad trust evaluation result for a services side might also lead to reduced trust in the user side identity management system.

- Structure the trust parameters visible on the second and third layers into the categories “Business reliability (comprising the parameter “blacklisted”) and “privacy” (comprising the parameters of security & privacy alert lists, privacy seals and PrimeLife function support). This structure should illustrate that the trust parameters used have different semantics and that scenarios with companies that are “blacklisted” for bad business practices, even though they have a privacy seal and/or support PrimeLife functions do not have to be contradictory, as they refer to different aspects of trustworthiness.
- Inform the user without unnecessary warnings - our previous usability tests showed that extensive warnings can be misleading and can even result in users losing their trust in the PrimeLife system. It is a very difficult task for the systems designer to find a good way of showing an appropriate level of alerting: for instance, if a web vendor lacks any kind of privacy seal, this in itself is not a reason for raising alarm, as most sites at present do not have any kind of trust sealing. We also did not choose the colour “yellow” for our trust meter for symbolising such an evaluation result that we called “fair” (i.e. we did not use the traffic light metaphor), as yellow already symbolises a state before an alarming “red” state.

### 13.3.3 Test Results

Usability tests for three iterations of our PrimeLife trust evaluation function mock-ups were performed in the Ozlab testing environment of Karlstad University in two rounds with ten test persons each and one round with 12 tests persons.

The positive results of the usability tests can be summarised as follows:

- Most participants seemed to understand the “Send Personal Data?” user interfaces and presented top-level trust evaluation results quiet easily. They thought that the UI was explicit and clear, with no distracting objects. The participants liked that the requested data were presented to them explicitly in “Send Personal Data?” before they decided to send their data or not.
- The “Good” and “Poor” emoticons on top level were also clearly understood by all users. Only the “Fair” emoticon (which we called “Not bad” and “OK” in the first two test rounds) was by some test participants interpreted as confusing.
- The colours red and green in the prototype (both on icons and over text) were all understood correctly by the participants.
- The icon for alarming the users was also correctly understood. This was not further evaluated in test 2 and 3.
- As many as 14 out of the 20 participants in test 1 and 2 liked the function they tested to be called “Trust Evaluation.”

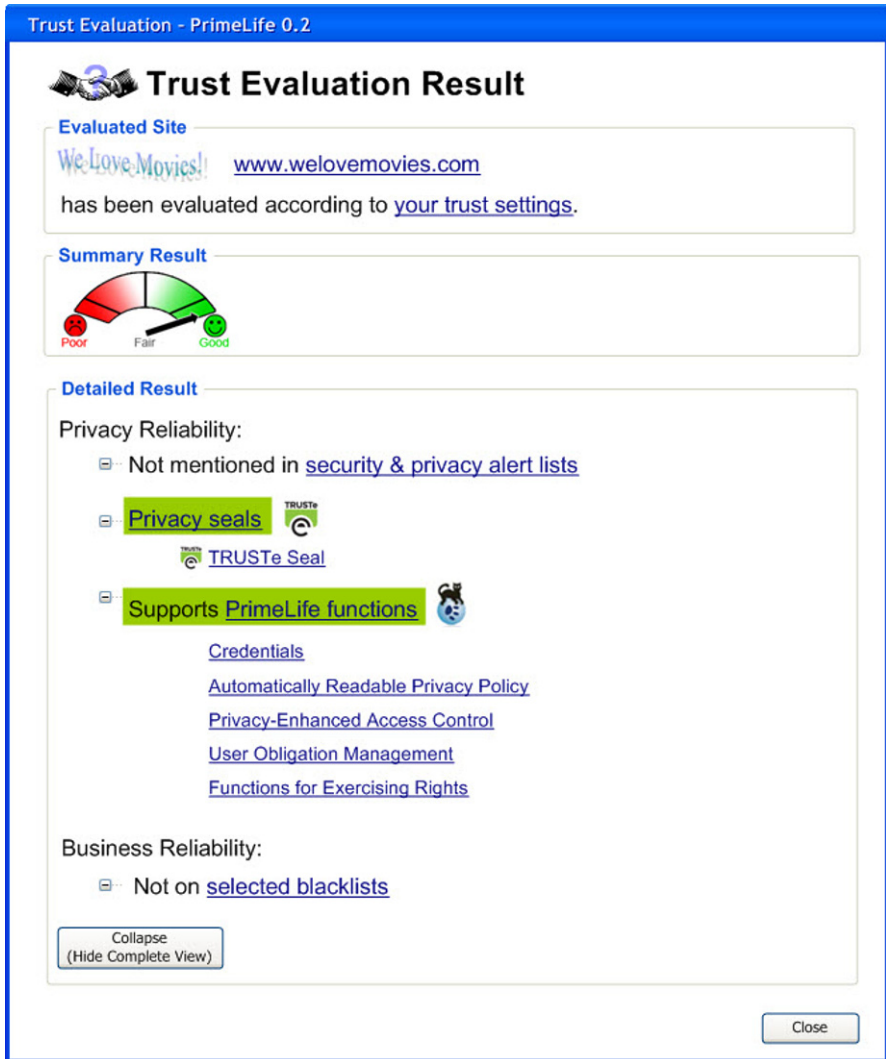


Fig. 13.3: Complete view of the Trust Evaluation Function (3rd Layer).

- All but one participant in Test 1 and 2 said in the interviews that they would like to use a PrimeLife prototype including a Trust Evaluation function that is similar to the one that was tested.
- Nearly all participants understood that the services side, and not the user side, was evaluated.

However, the tests also revealed a couple of usability issues that need to be addressed by our next iteration of mockups:

- The more detailed trust evaluation results on the second and third layer were harder to understand for most test persons. The most difficult evaluation result to interpret for the test participants was the detailed “Fair” evaluation result, several participants stated that the website was “not evaluated,” *“No evaluation is performed since everything is neutral”*.
- There has also been some confusion for some test users on how trust evaluation can work if the services side is not PrimeLife enabled. Hence, the users need to be better informed via the interface or other means that information about the services side’s trustworthiness can mostly be obtained from third parties and do not require PrimeLife support at the services side.

Moreover, it is interesting to note that some participants took a bad trust evaluation result on the parameter “Blacklisting” more serious than on “Privacy alerts.” One comment from an interview with a test person was: *“Alerts are warnings, but when you are blacklisted then it is really serious - this makes you think twice before sending my data.”*

## 13.4 The Data Track

The design of the Data Track is motivated by the finding that transparency and reversibility of transactions will enhance user trust (see above). The idea of the Data Track is to make it possible for users to get an overview of the data that they release in different web applications and under what conditions these data was released. The Data Track should also help the users in accessing remotely stored data about them and to make it possible to request or perform changes and deletion of this data. Thus, in essence the Data Track has two functions, a history function and a transparency function. The Data Track itself is in essence a viewer that draws information from three databases (see [Figure 13.4](#)). The different databases in turn store information on the PII (personally identifying information) sent, the sessions in which the data is sent and the changes that have been made to the data. The Data Track itself does not initially store the data in the PII and the session database when it is first sent, but rather relies on some other entity to do the storage. In our initial setup, the PRIME Core from the PRIME project stores the PII and session information when it is sent over the Web (see [Figure 13.4](#)). The Data Track does, however, modify the data and stores the changes to the data if the user requests changes and deletion. For a more detailed description please see [Pri10c].

As people engage in many transactions that involve multiple providers simultaneously, the implementation of a usable Data Track is difficult from an HCI perspective. Providing users with easy search tools for finding relevant records about past data disclosure is one example. Showing users that they can access their data on remote servers is another, which is especially difficult as users have little or no experience with this type of online access functions.

Within the scope of the PrimeLife project, we have conducted work for enhancing the functionality and usability of the Data Track. For this, we have specially de-



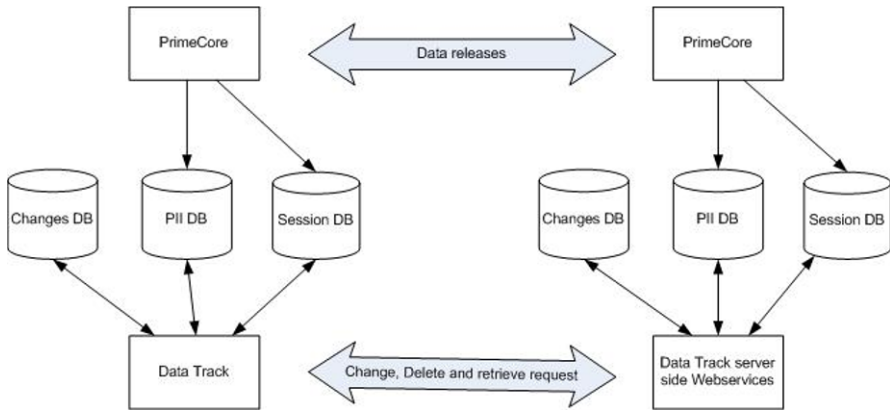


Fig. 13.4: Conceptual view of Data Track system as part of the PRIME core.

veloped and tested user-friendly search functionalities plus Online functions, which now allow end users to access, correct or delete their data at the remote services side (as far as permitted). For the development of a usable Data Track, we have followed an iterative design based on a cyclical process of UI prototyping, usability testing, and refinements of the Data Track user interfaces.

### 13.4.1 Use of the Data Track

When a user opens the Data Track, she will be confronted by a list of sessions in which she has released data. This view contains functionality for searching and filtering on specific data given in the sessions. Clicking on one of these sessions will open up a session window (see [Figure 13.5](#)). The session window will give the user information on what type of information, the actual value used, for what purpose and to whom the information was sent. By pressing the policy button (see [Figure 13.5](#)) the user will also get information on the agreed-upon privacy policy for this session. This window also gives the user the possibility to retrieve or delete remote information (if this has been permitted by the data controller). By pressing the “retrieve data from X” button, the user instructs the Data Track to contact the service’s web page through the “Data Track server side Webservices” and retrieve the data stored under this session in the Session and PII Database (DB) on the remote side (i.e the data that the data controller has on the user for this session) (see [Figure 13.4](#)). If the user presses the summary button, a Summary window will pop up (see [Figure 13.6](#)). This window contains similar information as the session window but displays all the information that the user has sent during all sessions with this specific data controller. The window also contains more fine-grained functions for deleting and changing specific values. In [Figure 13.6](#), the “Retrieve All Data from X” has already been

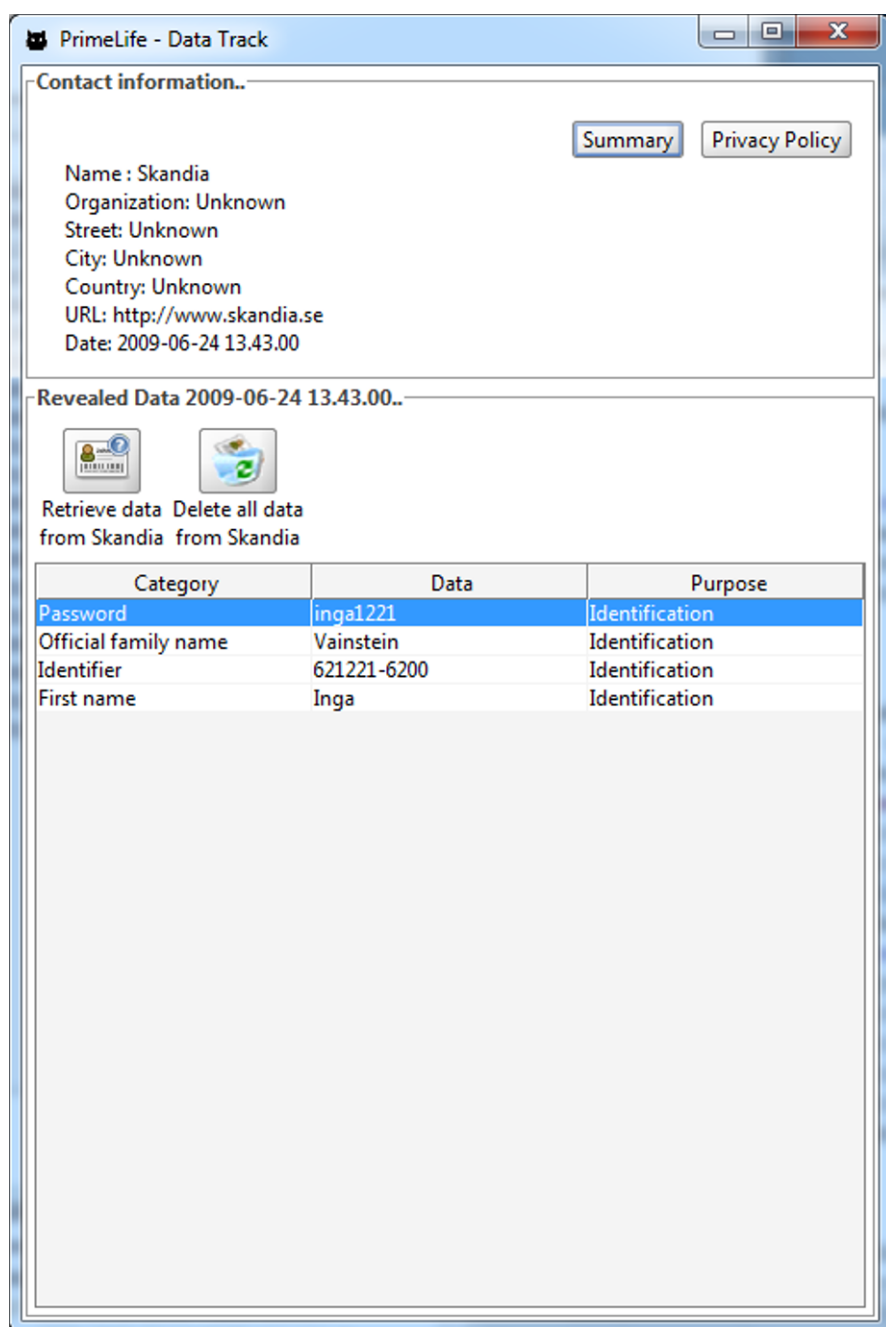


Fig. 13.5: Session Window in the Data Track.

pressed and the remote information is shown as well as the locally stored information. Any mismatch (such as different, missing or non-sent data stored on the remote side) between the sent and the remotely stored data will be marked, so that it can be easily spotted by the user (see [Figure 13.6](#)). This type of marking also takes place in the Session window but only for the data belonging to the same session.

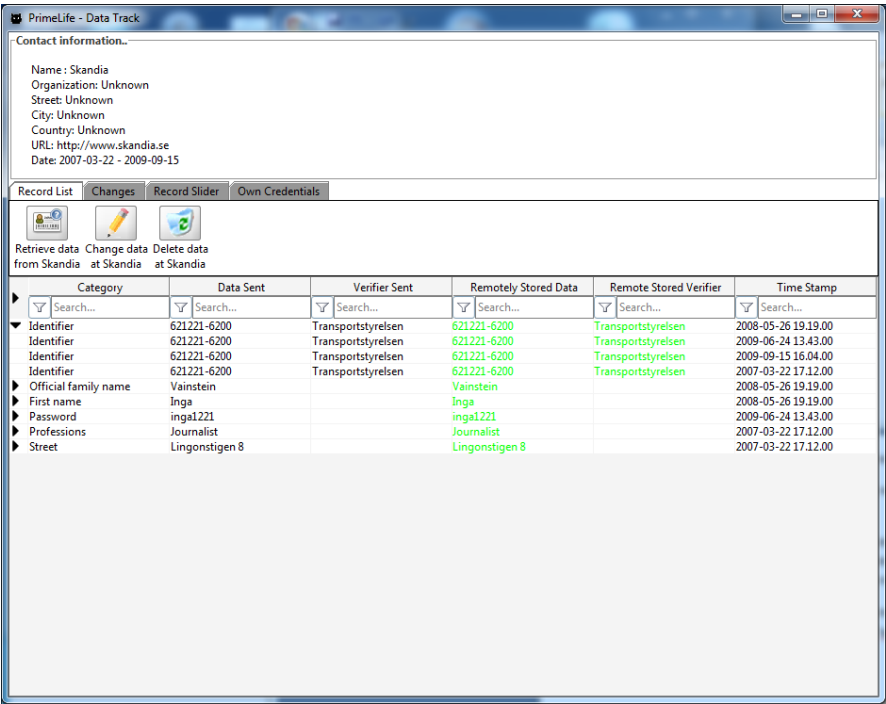


Fig. 13.6: Summary Window in the Data Track.

13.4.2 Test Scenarios & Test Setups

In total, we have performed 5 usability test rounds at Karlstad University and at CURE with a total of 58 participants involved. The test participants were aged between 19 and 56 years, 32 were male and 26 were female. All participants, except one, use the Internet on a daily basis. All participants shop online at least once a year. Most users stated they shop online once or several times a month and a few users stated they shop online once or several times a week.

The general purpose of all the tests was to evaluate the users’ comprehension of the Data Track UI. The first test was a pilot test to validate the test set up and

procedure, which was followed by three rounds of tests that differed only in regards to the amount of instructions given. The last test round was a combined test where the participants were asked first to use the Credential Selector UI (see Chapter 12) to perform a transaction followed by the Data Track test. This was done in order to see whether users would get a better grasp of the application if they got to experience both sending data and reviewing stored data first hand.

All tests followed the same procedure except in regards to the pre-test instructions. A test session lasted between 30 to 60 minutes and contained the following parts:

- Oral and written information about the test in general
- Pre-test questionnaire
- Pre-test introduction
- Pilot and first round of tests: instruction movie
- Second round of test: very short oral presentation after which users were given a few minutes to click around as a familiarising task
- Third round (Data Track and Credential Selection combination) as above but with additional information regarding the Credential Selection mock-ups.
- Test person reads task information and interacts with prototype
- In the third round (Data Track and Credential Selection combination), the participants were asked to use the Credential Selection UI to purchase a book from Amazon.com both before and after using the Data Track.
- Post-test questions
- Online Post-test PET-USES questionnaire
- Discussion about the given answers

During the test, users were asked to perform 15 different tasks<sup>1</sup>, which together would answer our questions regarding local search and online access functions.

More specifically we wanted to know if users would:

- Understand how to search within the tables?
- Understand how to add columns to the main table?
- See the sort function of the main tables?
- See the expand function of the tables?
- Understand how to open the “summary card”?
- Understand how to update information via the “summary card”?

### ***13.4.3 Results of the Usability Tests***

First of all, the results of the usability tests show that the usability of the Data Track is in a mature state. In general, most of the users succeeded in solving most of the tasks (see [Figure 13.7](#)).

---

<sup>1</sup> A more detailed description of the user tasks and the results of the tests can be found in [Pri10c].

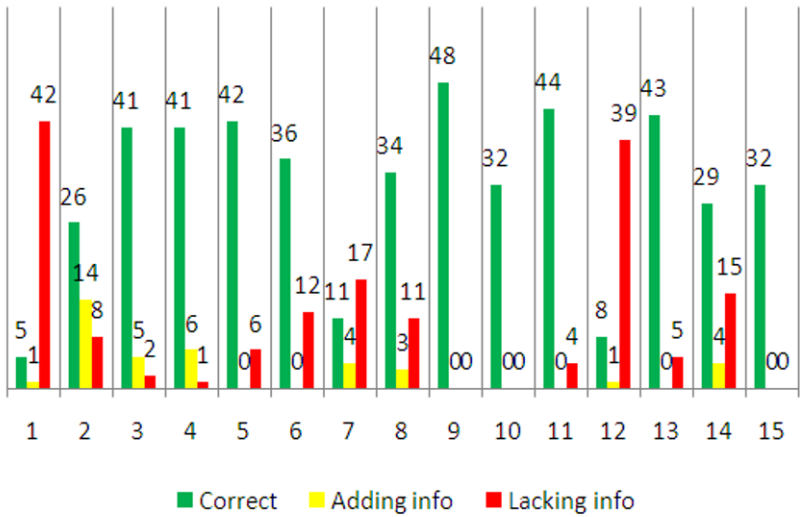


Fig. 13.7: The results of task 1-15 in regards to the participants completing the task correctly, adding information or giving less information than the correct answer required.

Having said this, it should be noted that users did not always accomplish this in an optimal way and that there is still, as always, room for improvement. Also, the tests showed a number of usability bugs that will not be reported here.

13.4.3.1 Results Regarding the Search Functionality

Given the fact that the idea is that the Data Track should be used for a very extensive period of time, users will aggregate an abundance of data. Hence, if users cannot successfully search through their data, the Data Track will not be of much use. The main issues with regards to the table UI was the interpretation of the summary line. Users either interpreted it as an occurrence, which made them overestimate the amount of times they had used a specific e-mail address, or missed the fact that it was expandable and thus instead underestimated their usage of a given piece of data. Other common mistakes were based on the understanding of the functionality of the columns heading, where some users did not understand that clicking the headers would sort the table and nearly no users realised that it was possible to add columns with specific information. The latter problem was leading users to open all summary cards and manually counting specific occurrences - something which is obviously not feasible after a couple of years of usage.

### **13.4.3.2 Results Regarding the Online Access Functionality**

The key features of the online access functionality are the possibility to remotely access and alter one's own personal data at the services' side. In order to do so, a test user was asked to retrieve remotely stored data, evaluate it, and proceed with correcting errors if such are found. A number of user errors during these tasks can be seen as spillover effects from other areas, such as users not correctly understanding the difference between the summary cards and transaction cards and users not understanding that the issuer of verified data is sent together with the verified personal information. However, the major issue of the online access functionality is that users do not clearly discriminate between data that is stored locally and remotely. Thus, instead of retrieving remotely stored data and evaluating it, the majority of users rather looked at a transaction card. Although this shows the user what data has been sent in a given transaction and gives the user the possibility to change data (e.g. in case of changed address), it does not let the user see a summary of all data aggregated by a service from multiple transactions and in the worst case from other sources.

### ***13.4.4 Discussion of Data Track Usability Tests***

On a general level, the results of the usability tests of the Data Track show that, with some exceptions, users have little trouble navigating the Data Track and finding information that is stored locally. Especially noteworthy is the use of the summary card, which all users understood correctly, and the table search function, which was also widely understood. The users' problems with the Data Track can be divided into two areas, namely UI problems and mental model problems.

With regards to UI problems, the main issue is the summary rows in the tables. The idea of the summary row is to show that the user has sent information to a given recipient. However, the problem is that users often do not understand that this is a summarising heading of possibly multiple attributes and that only the last value is being shown. This results in users not expanding the row and thus missing a lot of information that has been sent to the recipient. Quite the opposite has also occurred, namely that users have interpreted the summary row as a separate transaction making them overestimate the amount of data they have sent to the recipient.

With regards to issues based on users' mental models, the key problem is that users often do not distinguish between service and client side. This results in users not retrieving data from the service side in order to verify what information they have stored. Thus, tasks where the users can see the incorrect data locally have been satisfactorily solved, while tasks that depend on users retrieving remotely stored data have been more difficult.

Lastly, the combined tests did not show any reliable effect on users' understanding of the Data Track. However, three out of eight participants who overestimated the amount of data they had sent to Amazon with the Credential Selector actually

understood what they had actually sent after they had used the Data Track. Thus, using applications such as the Data Track and Credential Selector in combination helps users get into the right mental model.

## 13.5 Conclusions

In this chapter, we presented the HCI work on two different functions that we have developed within PrimeLife for enhancing reliable trust by end users, namely the trust evaluation function and the Data Track.

The tests of the trust evaluation function clearly showed that such a function is much appreciated by end users. The presentation of overall evaluation results on a top level, especially the green and red emoticons, as well as the fact that the services side was evaluated, were well understood. Some users had problems, however, understanding the “neutral” evaluation result (in case a side has no seal, does not support PrimeLife functions, is not blacklisted and does not appear on alert lists). Hence, the illustration of “neutral” results is one of the most difficult issues and still needs to be investigated further (see also [Pri09b]).

The results of the Data Track usability tests show that users have little trouble using most parts of the Data Track that concern locally stored data. With regards to locally stored data, it is mainly parts of the table UI that needs to be improved. A more challenging issue that needs further improvement is the problem of conveying to the users what is happening on the client side versus what is happening on the service side.

## Chapter 14

# HCI for Policy Display and Administration

Julio Angulo, Simone Fischer-Hübner, Tobias Pulls, and Ulrich König

**Abstract** The PrimeLife Policy Language (PPL) has the objective of helping end users make the data handling practices of data controllers more transparent, allowing them to make well-informed decisions about the release of personal data in exchange for services. In this chapter, we present our work on user interfaces for the PPL policy engine, which aims at displaying the core elements of a data controller's privacy policy in an easily understandable way as well as displaying how far it corresponds with the user's privacy preferences. We also show how privacy preference management can be simplified for end users.

### 14.1 Introduction

PrimeLife aims at developing privacy-enhancing identity management systems for technically enforcing user control and information self-determination. An important prerequisite for user control in the context of privacy-enhancing identity management are privacy policies, which can inform users about the personal data processing practices of a data controller at the time when she is requested to disclose personal data to that data controller. According to Art. 10 EU Data Protection Directive 95/46/EC (DPD), a privacy policy should inform a data subject at least about the identity of the data controller, the purposes for which the data are intended as well as any further information such as the categories of data and recipients concerned, her right of access to and the right to rectify her data, needed to guarantee fair personal data processing. However, in practice, privacy policies, whether posted on websites or contained within contractual texts, often include long complicated legal statements, which are usually neither read nor understood by the end users. Making privacy policies easily understandable and transparent is therefore an important challenge. In order to address this challenge, one emphasis of our HCI work within PrimeLife has been on privacy policy icons that support the display of a privacy policy through specially tailored graphical representations of policy aspects, which



will be discussed in the following chapter. Gross et al. [GA05] have shown that the perceived clarity of a privacy policy increases positive reaction to the site and its goals. Hence, easily comprehensible and transparent privacy policies are not only a means for enhancing user control, but can also serve the interests of the service providers.

Achieving better transparency of privacy policies is also the aim of privacy policy matching to allow users to make an informed decision on whether to use the service, as it is enabled by PPL (see Section 17.2.1 of this book). In this context, a user can state her privacy preferences to define under which conditions she would like to release what data. The user's preferences can be compared to the data controller's policy, so that the user can be informed in case her privacy preferences will not be met.

For ordinary users, defining and adapting privacy preferences in such a way that they protect their privacy properly are complex and error-prone tasks that usually require some expertise about basic legal privacy concepts and principles. In the non-electronic world, equivalent tasks do not directly exist, which means that ordinary users have no experience in defining and managing their privacy preferences. Without assistance, most users would very likely fail to define and use privacy preferences at all or could accidentally define or choose privacy preferences that are not as privacy-friendly as the users would like them to be. As security and privacy protection are often secondary goals for ordinary computer users [WT99], it is indeed not realistic to assume that users will spend much time and effort on privacy configurations. Hence, another major challenge, which we have addressed in PrimeLife, is the simplification of privacy preference management for end users.

For achieving this, our approach in PrimeLife has been to provide options of predefined "standard" privacy preferences, from which a user can choose and which she can customise "on the fly." If, for example, a data controller requests more data for a service than permitted by the user's current privacy preferences and the user agrees to it, the user will at the same time be asked whether she wants to adapt her preferences and possibly save them under a new name or whether she wants to overrule her preferences only for this single event. The set of predefined privacy preferences should represent the users' privacy interests and thus also includes the most privacy-friendly options for acting anonymously or for releasing as little information as needed for a certain service. For more advanced users, a preference editor is provided, which allows them, in a user-friendly way, to configure their individual privacy preferences.

For the development of our user interfaces (UIs), we have followed an iterative development approach with five iterations of UI development, testing and UI refinements and improvements. The emphasis of this chapter will be on the UIs developed within the fourth and fifth iteration cycles.

In this chapter, we present our work on user interfaces for the PPL policy engine, which aims at making privacy policies easily understandable and transparent for end users and at simplifying privacy preference management for them. The remainder of this chapter is structured as follows. Section 14.2 will discuss related work and refer to our previous HCI work in PrimeLife. In Section 14.3, we will present policy

management and display user interfaces for the PPL policy engine that we have developed in PrimeLife as part of the latest (4th and 5th) development iteration cycles. The results of usability tests of these UIs will be presented in Section 14.3.3 and conclusions will be drawn in 14.3.4.

## 14.2 Related Work

For making privacy policies and their core elements better understandable and more transparent, the Article 29 Data Protection Working Party has recommended providing policy information in a multi-layered format. Three layers of policy information are suggested: The short privacy notice (layer 1) must offer individuals the core information required under Art. 10 EU Directive 95/46/EC, which includes at least the identity of the controller and the purpose of data processing. In addition, a clear indication must be given as to how the individual can access additional information (of layers 2 and 3). The condensed notice (layer 2) includes all other relevant information required by Art. 10 of the Directive such as the recipients, whether replies to questions by the data controller are obligatory or voluntary, and information about the data subject's rights. The full notice (layer 3) includes in addition to layers 1 and 2 also "national legal requirements and specificities."

Our UI prototypes for policy display conform to the Art. 29 Working Party Recommendation.

Recent work on a "Nutrition Label" for privacy [KBCR09], has made proposals on how to present information to be displayed in short privacy notices in a user-friendly manner, namely the types of information to be collected, how this information is used and with whom it may be shared. In particular, a visualisation technique for displaying policies in a two-dimensional grid with "types of information" that are requested as rows and purposes as columns was developed and well received by test users. Our policy display UIs use a two-dimensional table presentation, which is similar to the proposed grid, for summarising what data is released to whom and for what purposes. We have, however, adapted it to meet PPL-specific requirements and have done further changes, which will be discussed in Section 14.3.2.1.

Moreover, there has also been previous work on the usability of P3P<sup>1</sup> user agents [CGA06], and the means for mediating information of P3P privacy policy compliance by websites to end users via the the Privacy Finder P3P-enabled search engine service [GECA06, TERS<sup>+</sup>06]. This related work on user interfaces for P3P and P3P-enabled tools does, however, not address requirements that can be derived from the EU privacy legislation. The Privacy Bird<sup>2</sup> is a P3P user agent that allows the user to specify her privacy preferences regarding a website's data handling policy. The privacy bird uses the traffic light metaphor for displaying information about the compliance of a site's policy with the user's preferences: If a site's policy meets

---

<sup>1</sup> Platform for Privacy Preference project, <http://www.w3.org/P3P/>

<sup>2</sup> <http://www.privacybird.org/>

the user's preferences, a small green bird icon in the browser's title bar emits a happy tweet after the page has been loaded. If the site violates the user's privacy preferences, the bird icon turns red with a shrill warning when the page is first loaded. For sites with no P3P policies, a yellow bird will appear. It is however questionable whether the traffic light is the right metaphor in this context, because having no privacy policy (symbolised by the yellow bird) can actually be regarded as worse than having a policy not matching the user's preferences (symbolised by the red bird). For allowing users to specify their privacy preferences, a set of three predefined groups of preferences is provided, which can be customised by the user during the installation process and via the privacy-bird menu. However, in contrast to the approach that we have taken, the privacy bird does not allow for changing privacy preference settings semi-automatically "on the fly."

Also, in contrast to the PrimeLife Policy Language (PPL), for which our user interfaces were designed, P3P has several functional restrictions, in particular it lacks support for obligations, support for downstream data sharing as well as support for anonymous credentials. Moreover, P3P has only a focus on one type of interactions (web pages, http).

Our previous HCI work in PrimeLife included three earlier development cycles of mockups for policy display and management, which were presented in PrimeLife Deliverables D4.3.1 [Pri08] and D4.3.2 [Pri10d]. The mockups of the first three iteration cycles showed several usability problems, which led to some improvements in the mockups of the fourth and fifth iterations cycles presented below. Besides, they were not fully compatible with the PPL specification, as the specification was only available in the 2nd project year.

As part of the third iteration cycle, a so-called PrimeLife Checkout Protototype (PLC) was developed, which included a "Data to Transfer" sidebar (DTT). It received very positive feedback when tested at Karlstad University, and will for this reason be briefly presented here (see [Figure 14.1](#)). The objective of DTT is to visualise a user-friendly overview of which data will be transferred to which data controllers for which purpose. The DTT consists of a box that displays a list of the data controllers involved in a transaction, each represented with an own section. Sections are separated by a horizontal line. Each data controller section begins with the data controller's name, which is followed by a list of the data fields and the respective data values that will be transferred in the respective transaction.

In addition, the purposes for the data storage/processing and the duration of data storage are displayed. At the end of each section, a hyperlink leads to the detailed privacy policy of the data controller in full text. The DTT has been implemented in JavaScript and is updated in real-time depending on what options were selected by the user. Besides the problem of not being PPL compliant, users easily got the impression that the PLC interface appeared as part of a webshop's side rather than as part of the user's side identity management system. This was another reason for starting with new designs in the 4th iteration cycles, which will be presented below.

**Data to Transfer**

---

**to Webshop (Purchase):**

**Address Line 1 :**  
Parcelstation 101

**Postcode :** 12345

**City :** Mycity

**Country :** EU

**E-mail :** John@doe.eu

Data will be processed and stored for the purpose of:

- Tax - 10 years

[Detailed Privacy Policy](#)

---

**to DHL (Shipping):**

**Address Line 1 :**  
Parcelstation 101

**Address Line 2 :**  
12345678

**Postcode :** 12345

**City :** Mycity

**Country :** EU

**E-mail :** John@doe.eu

Data will be processed and stored for the purpose of:

- Tax - 10 years
- Delivery - 7 days

[Detailed Privacy Policy](#)

---

Fig. 14.1: The “Data to Transfer” sidebar.

### 14.3 User Interfaces for Policy Management and Display

This section will present and discuss the user interfaces that we have developed for the PrimeLife Policy Language PPL, including UIs for selecting the user’s privacy preferences and for the “Send Data” Dialog which displays policy related information and allows users to customise their preferences “on the fly.” We have developed these UIs within the fourth and fifth iteration cycles of our iterative design approach for Policy Management and Display UIs.

### 14.3.1 *Selecting Privacy Preferences*

When users are browsing the Web, it is likely that they want to assign different levels of trust depending on what web site they are browsing or what they are doing. PPL allows the user to define privacy preferences that can specify what types of data they are willing to release under which conditions (in particular, for which purposes and - in future versions of PPL - for which retention period, etc.). As we cannot assume that ordinary users can easily define and manage their preferences, our approach in PrimeLife is to provide three predefined groups of privacy preferences (which we have called “Nearly Anonymous,” “Minimal Data,” and “Requested Data”), from which users can choose from when contacting a data controller, and which they can customised “on the fly”. In addition, the Privacy Tuner, which is a Policy Editor developed for PPL, allows the user to create further different customised preference groups.

At each point in time, the user has exactly one active group of privacy preferences selected. If the user has not actively selected any group, the predefined group “Nearly Anonymous,” which is the most privacy-friendly one, is selected by default. We refer to the user’s active group as the user’s active privacy preferences.

Our user interfaces allow for changing the active privacy preferences via a bookmark list, via a menu accessible from the location bar or via the “Send Data?” Dialog window. The latter will be presented in the next section.

Figure 14.2 shows the menu as part of the bookmarks list (“Favorites” list in Explorer), which can be used for selecting either one of the predefined or the customised preference group named “My Shopping.” Selecting a privacy preference group in the subfolder of a bookmark makes the browser go to the bookmark in question using the selected group as the user’s active privacy preferences. This approach is an advanced version of the bookmarks-based approach for selecting privacy preferences, which was first proposed in [PFHD<sup>+</sup>05, Pet05]. It is analogous to how folders of bookmarks are displayed, where the bookmark is treated as a folder.

In Figure 14.3 the icon of the user’s active privacy preferences is shown in the location bar. By clicking the icon the menu is shown, allowing the user to change his active privacy preferences. Note that the location bar approach allows the user to change privacy preferences when already on a webpage, while the bookmark based approach is limited to locations bookmarked by the user.

Since the main purpose of the user’s privacy preferences is to match these preferences with a data controller’s policy, the “Send Data?” Dialog described in Section 14.3.2 allows users to customise their settings “on the fly.”

These three different ways of selecting the active privacy preferences provides users with easy means for changing their preferences before going to a website, while browsing the site and as part of the process of disclosing data to the website. Selecting privacy preferences from the bookmarks menu would change the location of the browser and the active privacy preferences. The icon would then be updated in the location bar to reflect the user’s new choice. Any later change to the privacy preferences in the location bar will override any selection made when browsing to the current site from the bookmarks menu. In the same way, any change to privacy

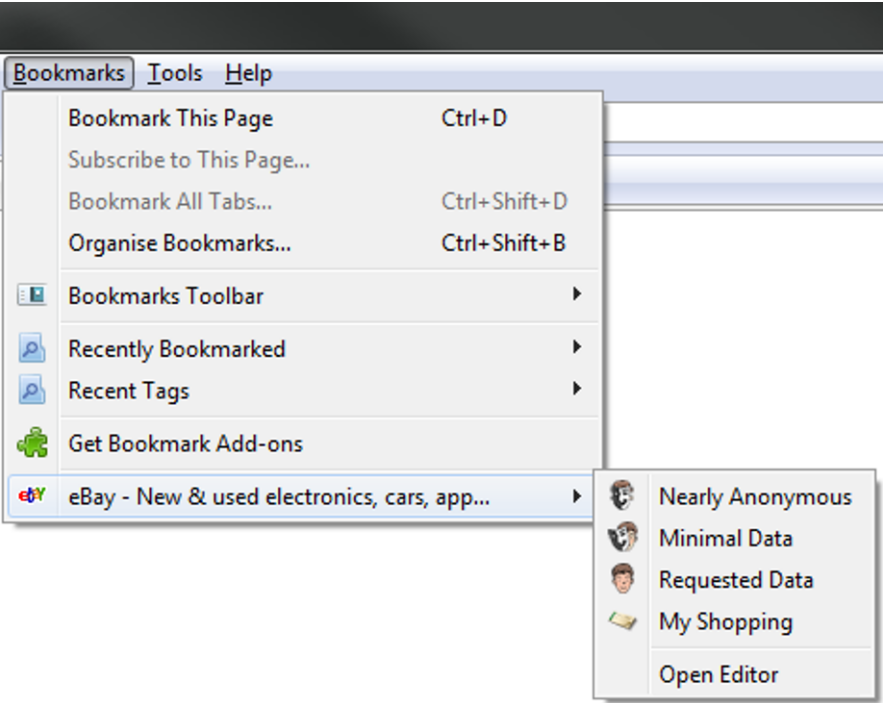


Fig. 14.2: The menu as part of the bookmarks list.

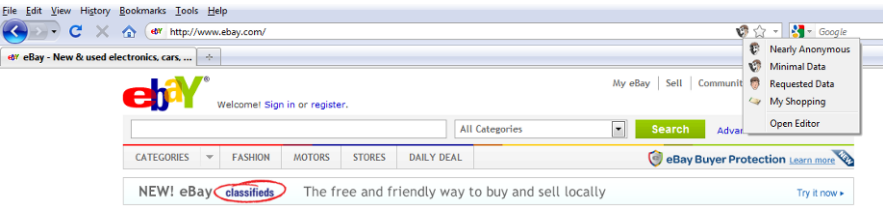


Fig. 14.3: Icon of the user’s active privacy preferences is shown in the location bar.

preferences in the “Send Data?” Dialog will override any selection made in the bookmarks menu or location bar. Trying to access the location bar with the “Send Data?” Dialog opened will result in the “Send Data?” Dialog being updated as well.

14.3.2 The “Send Data?” Dialog

The “Send Data?” Dialog appears in situations when a data controller requests a user to disclose personal data, for instance completing an online shopping transac-

tion. The dialog informs the user about the data controller's privacy policy and how well it matches the user's privacy preferences. It allows the user to either give her consent for sending personal data or to cancel the transaction. Moreover, it provides functions for the "on the fly" customisation of groups of preferences.

It should be noted that while the technical P3P and PPL vocabularies use the term "privacy preferences," we use in the "Send Data?" Dialog UIs the term "privacy settings" for the same concept, as we think that "preferences" is not the best term for conditions chosen or set by the user, which are not optional but that should be binding and can only be explicitly overruled by the user herself. Furthermore, tests of the comprehensibility of privacy terminology conducted by PrimeLife partner CURE showed that the term "privacy settings" was understood by the majority of test participants (5 out of 8), while privacy preferences was only understood by a minority (2 out of 14).

The "Send Data?" Dialog is one of the main user interfaces for the PPL engine and has to fulfill several PPL-specific requirements as well as legal and HCI-specific requirements, which are all listed below:

- Displaying the data controller's policy in an easily comprehensible manner as a basis for obtaining the user's informed consent to data disclosures. For achieving this, we are in particular following the Art. 29 Working Party's recommendation of displaying policies in multiple layers.
- Assisting the user in the process of selecting one combination of credentials that can be used to fulfill a data request by a data controller (as specified in a resource policy). The PPL engine provides the UI with all possible combinations of the user's credentials that can be used for the disclosure in question; it is up to the user, with the aid of the UI, to select which combination to use.
- Allowing the user to fill in attribute values for uncertified attributes. A resource policy can require both certified and uncertified attributes to be disclosed. For certified attributes, the values can be found in the credentials selected by the user. Uncertified attributes on the other hand can have any value the user wants to. A good example of a usually uncertified value is the user's email address.
- Providing the user with documentation and feedback information on the different aspects of the interface that will help clarify their intentions. Since the concept of online privacy is not simple to understand, it is at times necessary to assist the user in an unobtrusive manner.
- Making the user understand that the dialog handles information on the client's side and it is not part of a service provided by a data controller. During previous HCI work, we detected that users often have difficulties in differentiating the user side from the data controller's side [PFHD<sup>+</sup>05].
- Clearly displaying policy mismatches in an informative but not too alarming manner, so that users will make rational decisions on how to proceed. In case of a mismatch, the option to overrule the user's preferences for the current transaction only or for all future transactions shall be offered to the user. The second option requires the user to customise her current profile of privacy preferences accordingly "on the fly."

Moreover, PPL has the following properties that need to be taken into consideration:

- Personal data attributes requested for certain purposes cannot be marked as optional. Opt-ins for the use of attributes for certain purposes (e.g., use of an email address for marketing purposes) has to be done on the data controller's side before the PPL request for personal data is sent to the user. If the user opts in at a site, the data request from that site will include the attributes for the purposes that the user opted in.
- It is possible to specify that data will be also forwarded to a third party (downstream data controller), but the identity of that third party cannot be specified. If however, a site requests data, which should only be used by another site acting thus as the data controller, and if the data is therefore encrypted with a key of that data controller, and the site requesting the data forwards the encrypted data directly to that data controller, then it is possible to specify the identity of that data controller in PPL (e.g., if a web shop requests payment data encrypted with the public key of a payment provider, then the identity of the payment provider can be specified, who will act as a data controller).

#### 14.3.2.1 Meeting Requirements with the Design of the “Send Data?” Dialog

The list of legal, implementation-specific and HCI requirements given above determined to a great extent the development process and design decisions of the “Send Data?” Dialog. The user interface of the fifth iteration cycle, for which we conducted usability tests, is shown in [Figure 14.4](#).

The following paragraphs explain in closer detail how this design of the “Send Data?” Dialog meets the specified requirements.

The dialog is divided into an informative top panel and other three main panels conveying different privacy related information to the user. The purpose of the top panel, with the title ‘*Why am I seeing this dialog*’, is to describe the motivations of the dialog and how it can help the user protect her privacy.

The first main panel, with the title ‘*Data requested by*’ as shown in [Figure 14.4](#), lists all data controllers that are requesting some kind of information for a particular transaction. Along with the data controller's name, the user can find the data controller's contact information and a direct link to its full privacy policy (in order to fulfill the recommendation of Art. 29 Working Party). Data controllers are preceded with a circled number, such as ①, ②, ③, etc., in order to create a visual connection to their representation inside the two-dimensional table found under the second panel.

The second panel, titled ‘*Your data is requested for the following purposes*’, contains the table shown in [Figure 14.5](#) which corresponds to the simplified *Grid of a Privacy Nutrition Label* presented in [KBCR09]. This way of presenting privacy policies in the form of a *nutrition table* was encouraged due to the positive user understanding results reported by [KBCR09]. In the “Send Data?” Dialog, by looking at the *nutrition table* the user can perceive what types of personal data are requested



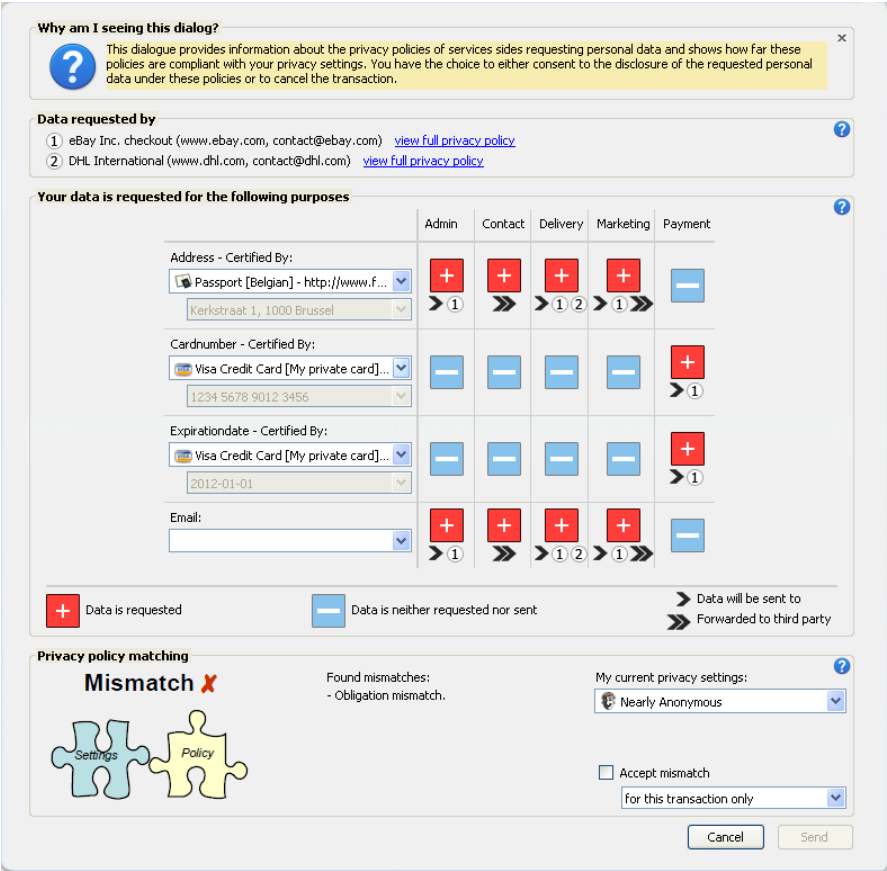


Fig. 14.4: The “Send Data?” Dialog.

in the table’s rows, and for what purposes these data will be used in the table’s columns.

There are several differences in the way the *nutrition table* is used in the “Send Data?” Dialog compared to the one suggested by [KBCR09]. First of all, the “Send Data?” Dialog allows displaying policies of more than one data controller requesting personal data. Our design of the “Send Data?” Dialog uses similar icons as in [KBCR09] inside each cell to indicate if data is requested for a specific purpose (+) or not (-). In addition, the “Send Data?” Dialog uses further icons as cell entries to show which data controller (represented by the circled numbers) is requesting the data (1, 2). The user is also able to see if the privacy policy specifies that her personal data will be forwarded to a third party for a specific purpose (1, 2), whereas the *nutrition table* presented in [KBCR09] only allows for specifying that data will be shared with third parties with the help of extra columns but without restricting the purposes for which the data may be shared or forwarded. Furthermore, in contrast to

[KBCR09], combo-box controls are used in our UIs to let the user select certifying credentials or freely type in uncertified credentials. Moreover, the graphical icons used to indicate whether data is being requested or not were also modified to better fit the purposes of the “Send Data?” Dialog. A more detailed description of these differences can be found in [Pri10d].

The use of this two-dimensional table aims at satisfying the requirement of providing the user with a more understandable summary of the data controllers’ privacy policies, while at the same time following Art. 29 Working Party’s recommendations of displaying only core policy information on the top layer of a multi-layered structured policy display.

The design of the credential selection for the “Send Data?” Dialog was based on a card-based approach as described in Chapter 11.5. In this approach a plastic card metaphor is used to represent the attributes that can be contained in, for example, a typical credit card or identity card. The requirement imposed by the disclosure of certified and uncertified attributes is fulfilled by employing different interface controls. By using textbox controls, the user is given the opportunity to enter any value for attributes that are uncertified. With the use of combo-box controls, the user is obliged to select a credential that certifies the attribute that is about to be sent and cannot be marked as optional. Furthermore, the user can see, but not edit, the values corresponding to the certified credentials that are displayed inside non-editable textboxes.

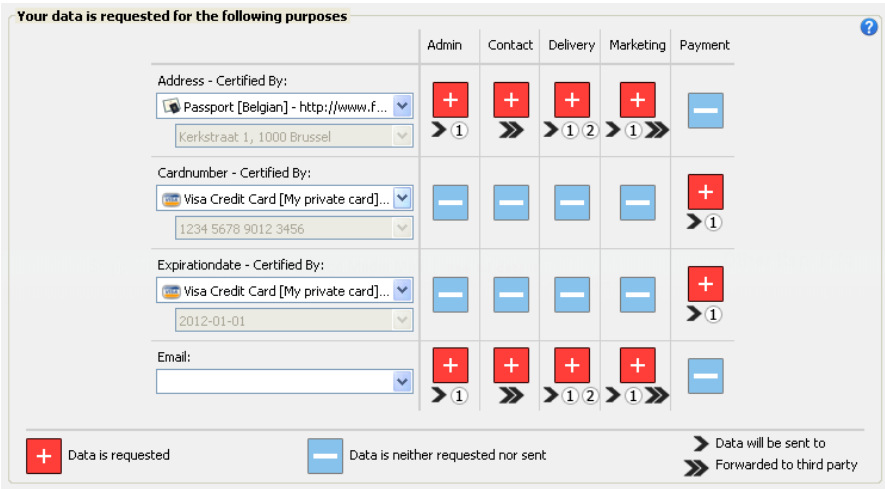


Fig. 14.5: The “Send Data?” Dialog – *Your data is requested for the following purposes.*

In the bottom panel, with the title ‘Privacy policy matching’, the user is given visual feedback on whether her privacy settings (i.e., her current preferences) match the data controller’s privacy policy or not, as shown in Figure 14.6. This feedback

is provided in the form of an indication icon based on a metaphor of two puzzle pieces fitting together, one representing the user's settings and the other one representing the data controller's privacy policy. If the pieces do not fit, it serves as an indication that the user's settings and the data controller's policy do not match. The icons are enhanced with text to clarify their meaning to the users, and are given the titles "Match ✓" or "Mismatch ✗," and each puzzle piece is named "Settings" and "Policy." Furthermore, the user is given a list of found mismatches and the reasons why a mismatch occurs. These images representing a "Match ✓" or "Mismatch ✗" fulfill the requirement of clearly displaying policy mismatches in a not too alarming manner, since the user is able to perceive that something is not right, but not shockingly wrong, while at the same time feedback is given on what is creating the mismatch.

At the same time, the user is given the possibility to adapt her current privacy settings as the transaction takes place, thus satisfying the requirement for "on the fly" privacy settings management. If a mismatch exists, the user is still able to proceed with the transaction if she consciously accepts the mismatch and explicitly acknowledges that her current privacy settings are overruled. Additionally, she is given the possibilities to overrule her settings for the current transaction only, to update her settings for future transactions, or to adapt her settings for future transactions and save them under a new name.

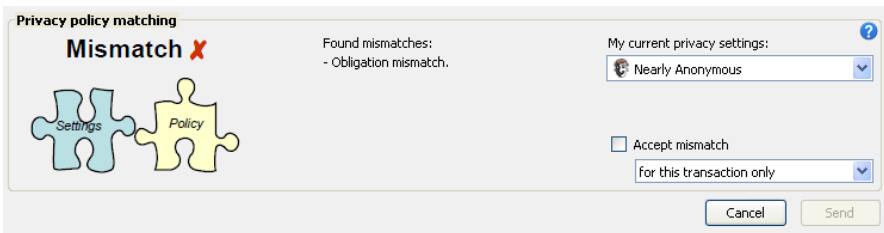


Fig. 14.6: The "Send Data?" Dialog – *Privacy policy Matching*.

Implementing the "Send Data?" Dialog as a Firefox 4 add-on has the advantage of forcing the dialog to 'pop-up' on top of the browsed page, due to the browser's new notification system. Therefore, when the "Send Data?" Dialog appears, the website behind it is dimmed, as shown in Figure 14.7, thus taking the focus of the user away from the website and placing it on the dialog. In this way users could have a clearer understanding that their information and privacy settings are being managed on their side and not on the data controller's side, since it is made clear to the user that the dialog belongs to the browser, which is installed locally.

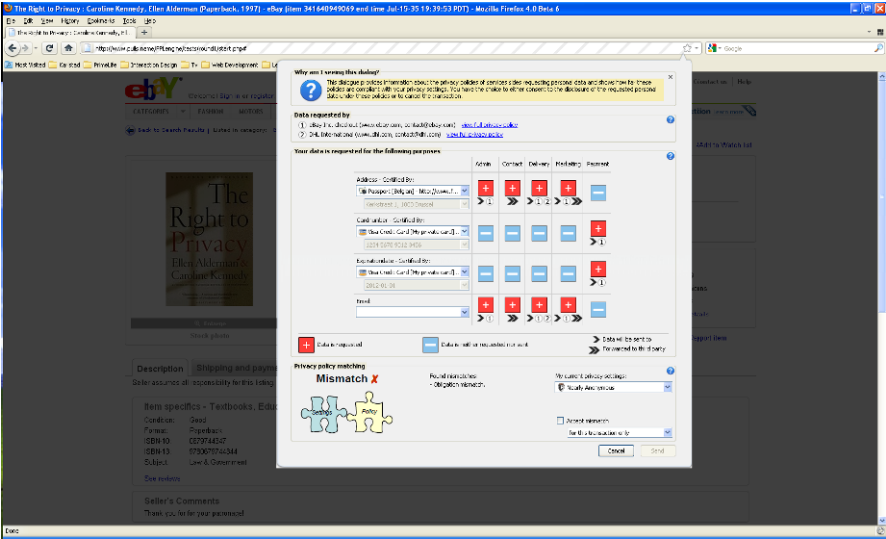






Fig. 14.7: The “Send Data?” Dialog – The website is dimmed as the dialog opens on top.

14.3.3 Testing the Usability of the “Send Data?” Dialog

As discussed earlier in this chapter, the concept of online privacy management and the adjustment of privacy settings is hard to understand for the average user, since there is no equivalent metaphor in an offline world. This created a challenge when carrying out usability tests for the proposed design of the “Send Data?” Dialog presented hereby. Special considerations were placed on the way test participants were introduced to the test sessions and the explanation they were given about the dialog before each session started.

Twelve test sessions were carried out at Karlstad University with participants coming from different occupations and social backgrounds (three coming from Asia, one from Africa, and eight from different European countries), out of which seven were females and five were males ranging from 19 to 34 years old. In each test session the participants were introduced to the concept of the “Send Data?” Dialog. They were shown the dialog for some time and were asked to respond to predefined interview questions while they interacted with the dialog. Notes were taken and, during some of the test sessions, the computer screen was recorded as well as the participant’s eye-gaze. At the end, the participants were asked to fill in a questionnaire containing selected PET-USES measurements, as presented in Section 10.4.

The general observations from the tests sessions and other expert evaluations, along with some suggestions for improvement, can be summarised in the following points:

- Participants in general understood that their information was not yet sent at the moment the “Send Data?” Dialog appeared, and that they had the opportunity of cancelling the transaction without compromising their information on the internet. What is more, participants understood that the program was not a service provided by the data controller but that all their information was handled locally, until the moment the *Send* button was pressed.
- Eye-tracking data showed that participants usually skipped the information text provided to them on the top panel. Many of them confessed that they either did not read it at all, that they read it but did not understand it, or that they read it only when they did not understand something else and were hoping to find answers on it. No participant tried to look for information on the question mark icons, , even when they were having trouble answering one of the interview questions. It has been suggested to remove the top panel for the next iterative cycle of the dialog.
- The test participants often failed to quickly recognize the connection between the circled numbers shown besides the names of the data controllers and the circled numbers in the two-dimensional table. Therefore, it was hard for the participants to understand who was requesting their data. However, after some time of familiarizing themselves with the interface, participants noticed this connection and most of them were able to assert which data controller was receiving which information. Representing data controllers with colors has been suggested as a way to make this connection more visible and understandable to the users. Also, placing the list of data controllers below the *nutrition table* and nearer to the other table’s legends might increase the visual perception and understanding of its relation.
- Most participants believed that the two-dimensional *nutrition table* represented the extent to which their privacy settings matched the data controller’s privacy policy. During the test sessions, participants tended to click on the red *plus* icon, , and they expected that the icon would change its looks into a more positive state. One participant even commented that “it [the icon] looks like a button.” Moreover, eye-tracking data suggests that the participants’ visual attention is placed on the table as soon as the dialog opens, which might be due to the table’s central placement and its contrasting red and blue colors. Expert evaluators have suggested removing these icons when data is being requested, since there is already a symbol indicating that data will be sent to a data controller, namely the arrows pointing to the circled numbers and the double arrow for forwarding data to a third party (i.e., , and place a simpler icon when data is not being requested at all (such as ).
- Participants had a very hard time understanding how some attributes of their personal information are certified with the use of credentials. It was hard for them to grasp that their information could be somehow stored in a computer and be ready to use for completing online transactions. In some cases, they even believed that their information was already stored somewhere on the Internet, and also that the data to be sent was the certifying credential itself and not the attribute.

- The puzzle metaphor represented by the image on the bottom panel, which indicates a *match* or a *mismatch* of privacy settings, was also not immediately clear to some test participants. Two of them thought at first that the image was a corporate logo or a piece of advertisement. However, when the “Mismatch ✖” image was pointed out, the majority of participants seemed to grasp the idea that there was *something* not right on the way the data controllers would use their information. One participant suggested that the puzzle “image should be located at the top” of the dialog and that it should be more visible when the dialog pops up. This suggestion was considered as relevant, since one of the main intentions of the user when seeing the dialog is to identify if her privacy preferences match or do not match the privacy policy of the data controllers, and the puzzle image is a graphical representation of this information. However, another important intention of the user is to distinguish which pieces of information are going to be sent on a particular transaction. Thus, placing the matching or mismatching elements at the bottom of the dialog persuades the user to verify the information to be sent in a transaction first and helps her make a more conscious decision.
- From the results of the PET-USES, it can be concluded that, in general, participants regarded the dialog as a useful tool to understand who would receive their information, to know what type of information they were releasing, and to make it easy to decide the amount of information to release for each transaction. Interestingly enough, the participants reported that they did not feel safe enough releasing personal information even when the dialog conveyed it was safe (i.e., two test participants *strongly disagreed*, three *disagreed*, and three *neither agree nor disagreed* with the PET-USES statement ‘*I feel safer realizing my information when the system states it’s ok*’). However, three participants *strongly agreed* and seven *agreed* with the statement ‘*I feel safer knowing that I will be notified by the system if I’m about to release more information than my chosen preference.*’

### 14.3.4 Conclusions and Outlook

In this section, we presented our work on user interfaces for the privacy policy engine and their evaluations.

In general, it is fair to say that most participants of the usability tests that we conducted understood the purpose of the program after some minutes of reflection and familiarisation with it. A couple of participants expressed their positive reactions towards the implementation of the “Send Data?” Dialog concept, arguing that they would be very interested in using such a program if it were fully implemented. The valuable feedback from these test sessions was used to improve the look-and-feel of the dialog and to make it more understandable for the average user, resulting in the design suggestion shown in [Figure 14.8](#).

Feedback has also been collected from initial usability tests of this last redesign. Results from the latest round of testing revealed a general improvement in the usability and general user understanding of the “Send Data?” Dialog prototype. For

Send Data?

Your data will be sent and used for the following purposes

	Admin	Contact	Feedback	Marketing	Payment
<div>Name - Certified By:</div> <div> <div>Driver's License [Swedish] - ...</div> <div>Inga Vainstein</div> </div>	> 1	>>	> 1	> 1 >>	-
<div>Cardnumber - Certified By:</div> <div> <div>Visa Credit Card [My private...]</div> <div>1234 5678 9012 3456</div> </div>	-	-	-	-	> 1 2
<div>Expirationdate - Certified By:</div> <div> <div>Visa Credit Card [My private...]</div> <div>2012-01-01</div> </div>	-	-	-	-	> 1 2
<div>Email:</div> <div></div>	> 1	>>	> 1	> 1 >>	-

> Data will be sent to:

1 eBay Inc. checkout (www.ebay.com, contact@ebay.com) [Privacy Policy](#)

2 Visa (www.visa.com, customersupport@visa.com) [Privacy Policy](#)

>> Data will be forwarded to others

- Data will not be sent

Privacy policy matching

Settings

Policy

Your [Privacy Settings](#) do not match with 1's [Privacy Policy](#).

Found mismatches:

- You do not allow your Email to be used for Marketing
- You do not allow your Email to be forwarded to others for Marketing

My current privacy settings:

Minimal Data

☐ Accept mismatch
 

for this transaction only

Cancel

Send

Fig. 14.8: The “Send Data?” Dialog – New redesign after the feedback obtained from the current iteration cycle.

example, participants found it easier to understand which attributes of the certifying credentials were about to be sent to the data controllers. Also, by removing the red and blue icons from the table, participants did not focus most of their attention on the table and other elements of the interface were more noticeable. Participants demonstrated having a good understanding of how to change their privacy settings and of the concept of “on the fly” privacy management. Most participants also expressed a better acceptance of the icons and images used. More importantly, participants had a better understanding of what a *mismatch* consisted of and how the program could help them protect their privacy. On the downside some of these findings suggest that users still think that the *nutrition table* representing the purposes for which data will be used is a representation of their own privacy settings. Participants seemed to still think that the table shows the degree to which their privacy settings match the data being requested by the data controller.

Further work will continue to investigate the additional improvement in the “Send Data?” Dialog’s user interface. For example, the possible use of subtle icons to represent the types of credentials listed in the rows, replacing the circled numbers by the data controller’s corporate logos to provide the user with stronger visual cues, as well as other visual and interaction enhancements. We might also extend the compatibility of the “Send Data?” Dialog to be used with other web browsers.

Furthermore, integrating the privacy editor in which users can modify their privacy settings is also considered for future work.





# Chapter 15

## Privacy Policy Icons

Leif-Erik Holtz, Harald Zwingelberg, and Marit Hansen

**Abstract** Many individuals are not aware of who is collecting and handling their personal data for what purpose. Usually privacy policies are too long, too complicated to understand, and reading them is hardly appealing. To improve the awareness and comprehension of individuals on what is happening with their personal data, privacy icons are being proposed. The PrimeLife project has developed icon sets for different use cases such as e-commerce, social networks and handling of e-mails. It conducted user tests and an online survey to analyse how well users understand what the privacy icons should express. This section summarises the findings of PrimeLife's work on privacy icons.

### 15.1 Introduction

Content that should be quickly understood by a broad audience is often expressed via icons, e.g., symbols pointing to fire exit or subway station. In general, well designed icons are able to convey information by means of one single graphical representation expressing the relevant content in a manner understandable for wide audiences, ideally even across cultural domains. This leads to the idea of developing icons that inform data subjects about privacy-relevant issues. In particular, parts of privacy policies could be expressed by icons making the content of these legal documents easier to access and comprehend. Other use cases for icons within the privacy area intend to display privacy aspects and implications of user actions in the user client, on websites or in social networks. Within the PrimeLife project, this idea has been taken further, scenarios for the deployment have been developed, and sets of icons were tested within usability tests.

This section elaborates on the motivation for the approach chosen by PrimeLife, describes the developed icon sets and presents results from a user test and an online survey. While these icon sets deal with services offered by data controllers who are the addressees for incorporating icons in their service, a different area of use can

be a peer-to-peer scenario when users themselves attach to their data information on how they want others to handle these data. An example for this approach for the scenario of sending and receiving e-mails is sketched at the end of this section. Finally the conclusion and outlook section summarises the findings and further steps to be taken for research and development as well as policy makers.

## 15.2 Motivation for Introducing Privacy Icons

The potential area of application for privacy icons is broad [Han09]. Icons may be deployed for indicating rights and limitations for own data provided via e-mail, for social networks, blogs, or websites showing prominently an illustrated abstract of their privacy policy. Machine-readable policies, as provided by some websites, may also be interpreted and translated into icons on the client side. Also third-party services commenting others' privacy policies, e.g., [GPS09], may deploy icons in their implementations.

However, it should be clear the use of icons cannot and does not intend to replace fully written privacy policies as the basis for informed consent, according to European Privacy regulations. But privacy icons may be used to supplement written privacy policies, pointing to relevant sections, e.g., by adding them as an initial to a paragraph [Pri08, p. 28]. Icons may also serve as a very abstract but easy to access level of layered privacy policies as has been suggested by Art. 29 Working Party [Par04].

Within the PrimeLife project, a large icon set has been developed with the aim to test the user's understanding in respect to more complex or less known principles. The research findings will aid in the development of understandable icons sets for use in specific environments.

## 15.3 Related Work

The idea of expressing relevant statements from privacy policies using icons had been developed earlier within the privacy community. To our knowledge, Mary Rundle was the first to propose a draft set introducing icons for privacy statements in a Creative Commons-like style [Run06]. Her approach offers a brief set of icons for different purposes and data types and was meant to foster further discussion on this topic. The icons do not take the existence of a legal data protection framework for granted, but rather build upon the U.S. understanding of privacy [Han09]. Matthias Mehldau independently developed an icon set, also inspired by the Creative Commons licenses [Meh07]. Based on European understanding of data protection, his approach does not only depict data types and groups of recipients but also includes icons for certain purposes. Other proposals for introducing a graphical representation of privacy statements or privacy properties were made by Bickerstaff [Bic09],

Kelley et al. [KBCR09], Helton [Hel09], Raskin [Ras10] and in the KnowPrivacy report [GPS09]. In the area of social networks, Iannella and Finden have made the attempt to introduce a set of icons on access rights from peers to the user's data. These icons are accompanied by machine-readable statements in a policy language [IF10]. For a detailed introduction and references to further suggestions and analysis see the related PrimeLife Deliverables D4.3.1 [Pri08] and D4.3.2 [Pri10d].

## 15.4 PrimeLife Icon Sets

So far, two icon sets have been developed within PrimeLife: one for general use and a specific set limited to show which user group is supposed to gain access to one's profile in a social network. Note that the proposed icons are not meant to be warning symbols for potentially risky processing, but rather aim at describing in a neutral way what happens with personal data. Some types of processing that are usually understood to be harmful may in some cases even be intended by a data subject, e.g., person-related profiling of interests can be of major value in a dating service to find the person sharing these interests.

This section firstly introduces the main characteristics of the different icon sets and then presents first results of user tests and an online survey for the evaluation of these icons.

### 15.4.1 *PrimeLife Icon Set for General Usage*

The icon set for general use contains the sub-categories data types, groups of recipients and processing steps including references to common purposes. Icons depicting data types include: personal data, sensitive data, medical data, payment data [Pri10d, p. 39]. Icons referring to specific types of processing and purposes comprise: legal obligations, shipping, user tracking, profiling, storage, deletion, pseudonymisation, anonymisation, disclosure to third parties and data collection from third parties [Pri10d, p. 39]. This icon set is suitable for being used in e-commerce scenarios. As a typical example, it contains a specific icon that expresses how long the user's IP address is stored. Already, the existence of such an icon may create awareness that even the storage duration of IP addresses can be relevant for compliance with data protection laws both on the service's and on the user's side. The use of icons on their own does not always achieve full clarity on the user's side about the intended data processing: For instance, the purpose "legal obligations" reflects the claim of the data controller that there is some regulation that demands data processing performed in the service's workflow, but does not go into detail. This information should be given at least in the full privacy policy of the data controller.

### 15.4.2 PrimeLife Icon Set for Social Networks

In social networks, further privacy-related statements are helpful for users, in particular to visualise who, usually in addition to the social network provider itself, will get access to which information. Therefore, the icon set for social networks contains icons for the following groups of recipients: selected individuals, friends, friends of friends and the whole network respectively the general public. The icons from the social network icon set may be used for depicting the personal configuration of privacy settings or the audience selection of individual pieces of content, e.g., to directly choose (groups of) individuals that may or must not gain access to selected data. In addition, the icons may work as reminder whenever the user looks at her profile. They can also be used in combination with components of the icon set for general usage.

## 15.5 Test Results

The icon sets developed in the PrimeLife project have been tested. For this purpose, the test group was asked to evaluate alternative icons that should express the same aspects to see which of the developed icons fits best. In this section, we describe important results and show the icons that will be used for the final icons set(s). An earlier test revealed shortcomings when displaying a large variety of purposes for different use cases by too many icons. This led to the development of one icon set for general use and an additional icon set for usage within social networks. These sets have been tested with 17 Swedish and Chinese Students each and in an online survey with 70 test persons. Participants of the online survey assessed themselves as being privacy-aware, the students were unaware. The test set ups and first results are described in [Pri10d]. Final test results and interpretations will be published in PrimeLife's final HCI report [Pri11].

The icons for address data, medical data, payment data, for the purpose shipping and for the data processing procedures storage and deletion (see [Figure 15.1](#)) were voted to be understandable, clear and helpful by the students, both Swedish and Chinese. In addition, these icons were also rated better than alternatives in a web-based survey.



Fig. 15.1: Excerpt of well understood icons for general usage tested by KAU.

Survey participants were also asked to express suggestions and comments. The floppy disk symbol raised the concern that it relates to an outdated medium and might become unfamiliar and hard to recognise in the future.

The online survey provided the participants with a scenario for each icon and asked how well an icon fit the described purpose or, in case of alternative icons, which of them fits best. Some of the icons had been found to be intuitive and easy to understand. These icons received the majority of votes as well as positive comments and included the representation of payment data, the icon for storage periods, and the icon stating that information will only be transferred to individuals selected by the user (see [Figure 15.2](#)).



Fig. 15.2: Excerpt of evaluated icons.

The test of the icons for visualising the recipients in social networks indicated that research is on the right track. The survey showed that the participants found the icons shown in [Figure 15.3](#) to best fit in order to depict the following groups of recipients: selected individuals, friends, friends of friends and the whole network.



Fig. 15.3: Excerpt of icons showing different groups of recipients within social networks.

The test results suggest that clear icons with few details are preferred. For the more complex concept “friends of friends”, the icon from the initial PrimeLife draft [Pri08] was chosen where the circles indicating friendship were seemingly clearer. Of course, a unification of the design style would be necessary before proposing these icons for a wider use.

For PrimeLife’s further research this raises the question: to what extent can and should more complex concepts be visualised by icons? A possible solution may be to develop sets of icons for specific environments and use cases strictly limited to a handful of easy to grasp icons as been suggested for the use in e-mail with the Privicons (see Section 15.6).

## 15.6 An Approach for Handling E-mail Data: Privicons

“Privicons” is the name of an approach developed by researchers of Stanford University, the PrimeLife project and interested individuals to convey the information about how the data in e-mails should be handled [Pro10]. By the Privicons approach, the sender of an e-mail message has a means to express her preferences on how the message content should be handled by the receiving user(s). For this purpose, the semantics of six icons in a graphical as well as in pure ASCII representation (“Privicons”) are described (cf. [Figure 15.4](#) that shows both kinds of representation for the six proposed Privicons). These can syntactically be integrated either in the first line of the body, in the subject line and/or in a dedicated header of any e-mail message. E-mail user agents can be implemented that support the users in handling the messages according to the Privicons statements given in [Figure 15.4](#).

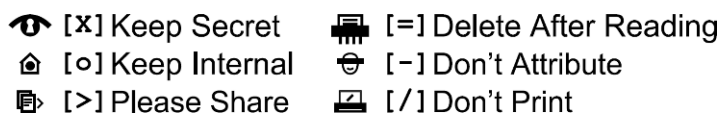


Fig. 15.4: Privicons.

By using the “keep secret” Privicon, the sender of the e-mail requests the recipient(s) to keep the received message secret. Related is the usage of the “keep internal” Privicon by which the receiving users are asked to present the e-mail message only to those people that are common friends, or otherwise qualify as “internal”, e.g., by being part of a group of people that are in a tight relation to both the sending user and the respective receiving user.

In contrast, the Privicon “please share” depicts the sender’s request or offer to the recipients to share the e-mail.

Again, another confidentiality-related Privicons is “delete after reading”: The sender requests the recipient(s) to delete the e-mail after reading it. The “don’t attribute” Privicon addresses information about the sender, i.e., it asks the receiving user(s) to not attribute, name or mention the original sending user of the e-mail message in any kind. If not stated otherwise, the receiving user(s) may quote, follow or paraphrase the content, facts and opinions voiced in the original e-mail message. Finally, the Privicon “don’t print” is self-explaining.

Meanwhile the Privicons project has drafted a Request for Comment to initiate the debate on the icons as well as how to embed them in day-to-day e-mail transfer. In this respect, the project is also working on proof-of-concept implementations.

## 15.7 Conclusions and Outlook

The results of the tests performed by the PrimeLife partners KAU and CURE indicate the further proceeding in the icons development. The well-rated icons will be refined to a final version of one or several smaller use-case specific sets. For this, the test results will be analysed *inter alia* to answer the question of how complex the depicted concepts may be. In addition, the lesson learned so far suggests that we stick with clear and simple icons.

The vast variety of research groups that work on some kind of privacy icons emphasises the need for standardisation. In parallel to standardisation efforts that, among others, should involve data protection authorities as well as user organisations, the approach of machine-readable privacy statements should be brought forward. The use of icons and the incorporation of machine-readable policies, as well as their relationship to each other and to today's practice of presenting privacy policies in legalese on the service's website has to be spelled out. For instance, it should be avoided that users looking at the icons get a totally different picture of the intended data processing than those reading the privacy policy in natural language or those who rely on the interpretation of the machine-readable policy by their user client. Further, thought should be given to incentives for data controllers to inform the data subjects in a better way than pointing them to the privacy policy and to educate individuals for better understanding of all aspects that are relevant to their privacy.





## References Part III

- [ACC<sup>+</sup>ce] C. Andersson, J. Camenisch, S. Crane, S. Fischer-Hübner, R. Leenes, S. Pearson, J.S. Pettersson, and D. Sommer. Trust in PRIME. In *Proceedings of the 5th IEEE Int. Symposium on Signal Processing and IT*, December 18-21, 2005, Athens, Greece.
- [Bic09] R. Bickerstaff. Towards a commons approach to online privacy a "privacy commons". Presentation at SCL Information Governance Conference 2008, London 2008 Updated presentation: Towards a Commons Approach to Online Privacy for Social Networking Services a "Privacy Commons" [http://www.ico.gov.uk/upload/documents/pio\\\_conference\\\_2009/roger\\\_bickerstaff\\\_birdandbird\\\_presentation.pdf](http://www.ico.gov.uk/upload/documents/pio\_conference\_2009/roger\_bickerstaff\_birdandbird\_presentation.pdf), 2008-2009.
- [Bra99] S. Brands. *Rethinking Public Key Infrastructure and Digital certificates – Building in Privacy*. PhD thesis, Eindhoven. Institute of Technology, 1999.
- [Bro96] J. Brooke. SUS: a "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, editors, *Usability Evaluation in Industry*. B. A. Weerdmeester and A. L. McClelland, 1996.
- [CGA06] Lorrie Faith Cranor, Praveen Guduru, and Manjula Arjula. User interfaces for privacy agents. *ACM Trans. Computer-Human Interaction*, 13(2):135–178, 2006.
- [Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.
- [Coo95] Alan Cooper. *About Face – The Essentials of User Interface Design*. Foster City CA: IDG Books Worldwide., 1995.
- [CSS<sup>+</sup>05] J. Camenisch, A. Shelat, D. Sommer, S. Fischer-Hübner, M. Hansen, H. Krasemann, G. Lacoste, R. Leenes, and J. Tseng. Privacy and Identity Management for Everyone. In *Proceedings of the First ACM Workshop on Digital Identity Management. DIM'05. ACM. November 11, Fairfax, Virginia, USA*, 2005.
- [CSSZ06] J. Camenisch, A. Shelat, D. Sommer, and R. Zimmermann. Securing user in-puts for the web. In *Proceedings of the Second ACM Workshop on Digital Identity Management. DIM'06. ACM, New York, NY*, 33-44, 2006.
- [Dir95] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, 23.11.1995.
- [FBL04] Ronnie Fanguy, Kleen Betty, and Soule Lori. Privacy policies: Cloze test reveals readability concerns. *Issues in Information Systems*, V:117–123, 2004.

- [FHFL09] Simone Fischer-Hübner, Steven Furnell, and Costas Lambrinoudakis. Exploring Trust, Security and Privacy in Digital Business. *Transactions on Large Scale Data and Knowledge Centered Systems*, 1(1), September 2009.
- [FHH00] E. Frøkjær, M. Hertzum, and K. Hornbæk. Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems*; 1-6 April 2000; New York: ACM Press, 2000.
- [GA05] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80, New York, NY, USA, 2005. ACM.
- [GECA06] J. Gideon, S. Egelman, L. Cranor, and A. Acquisti. Power Strips, Propylactis, and Privacy, Oh My! In *Proceedings of the Symposium of Usable Privacy and Security (SOUPS 2006)*. ACM Digital Library, July 14-16 2006.
- [Ger09] German Federal Data Protection Act, 1 January 2002, revised in 2009.
- [GKW<sup>+</sup>10] Cornelia Graf, Katrin Kristjansdottir, Peter Wolkertorfer, Claudia Oppenauer-Meerskraut, and Manfred Tscheligi. User understanding of data storage and data travelling, 2010.
- [GPS09] J. Gomez, T. Pinnick, and A. Soltani. Know Privacy. [http://www.knowprivacy.org/report/KnowPrivacy\\_Final\\_Report.pdf](http://www.knowprivacy.org/report/KnowPrivacy_Final_Report.pdf)\-therein: PolicyCodingMethodology:\url{<http://www.knowprivacy.org/policies/methodology.html>}, 2009.
- [GWGT10] Cornelia Graf, Peter Wolkertorfer, Arjan Geven, and Manfred Tscheligi. A pattern collection for privacy enhancing technology. In *PATTERNS 2010, The Second International Conferences on Pervasive Patterns and Applications*, pages 72–77, 2010.
- [GWKT10] Cornelia Graf, Peter Wolkertorfer, Katrin Kristjansdottir, and Manfred Tscheligi. What is your privacy preference? an insight into users understanding of privacy terms, 2010.
- [Han09] M. Hansen, editor. *Putting Privacy Pictograms into Practice – A European Perspective*, Proceedings of Informatik 2009 – Im Focus das Leben, 2009. Fischer, S. and Maehle, E. and Reischuk, R. (eds.). pp. 1703-1716.
- [HBK03] M. Hassenzahl, M. Burmester, and F. Koller. AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In J. Ziegler and G. Szwillus, editors, *Mensch & Computer 2003: Interaktion in Bewegung*, 2003.
- [Hel09] A. Helton. A Privacy Commons Icon Set. <http://aaronhelton.wordpress.com/2009/02/20/privacy-commons-icon-set/>, 2009.
- [HHN10] Leif-Erik Holtz, Marit Hansen, and Katharina Nocun. Towards displaying privacy information with icons. In PrimeLife Consortium, editor, *Proceedings of IFIP/PrimeLife Summer School*. Springer, 2010.
- [Hor06] K. Hornbæk. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies*, 64:79–102, 2006.
- [IF10] R. Ianella and A. Finden. Privacy Awareness: Icons and Expression for Social Networks. 8th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods incorporating the 6th International ODRL Workshop, Namur/Belgium, 2010.
- [Iiv04] Netta Iivari. Enculturation of user involvement in software development organizations - an interpretive case study in the product development context. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction*, NordiCHI '04, pages 287–296, New York, NY, USA, 2004. ACM.
- [ISO88] ISO. Ergonomic requirements for office work with visual display terminals (VDTs)-Part 11: guidance on usability-Part 11 (ISO 9241-11:1998), 1988.
- [KBKR09] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W. Reeder. A “Nutrition Label” for Privacy. In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, New York, NY, USA, 2009. ACM.
- [KWGT09] Christina Köffel, Peter Wolkertorfer, Arjan Geven, and Manfred Tscheligi. A study on dynamic vs. static display of privacy preferences, 2009.

- [KWW08] Christina Köffel, Erik Wästlund, and Peter Wolkerstorfer. PRIME IPv3 Usability Test Report, July 2008.
- [LCP06] H. Laché, S. Crane, and A. Pippen. Trustguide: Final report. Technical report, HP Labs, October 2006.
- [LLPH05] R. Leenes, M. Lips, R. Poels, and M. Hoogwout. User aspects of Privacy and Identity Management in Online Environments: towards a theoretical model of social factors, in PRIME Framework V1 (chapter 9). Technical report, PRIME Project, June 2005.
- [Meh07] M. Mehlau. Iconset for dataprivacy declarations v0.1, 2007.
- [Nie00] J. Nielsen. Why you only need to test with 5 users. [www.useit.com/alertbox/20000319.html](http://www.useit.com/alertbox/20000319.html), 2000.
- [Nor88] Donald Norman. *The Design of Everyday Things*. New York: Double-day/Currency, 1988.
- [NW06] David G. Novick and Karen Ward. Why don't people read the manual? In *Proceedings of the 24th annual ACM international conference on Design of communication*, SIGDOC '06, pages 11–18, New York, NY, USA, 2006. ACM.
- [Par04] Art. 29 Working Party. Opinion 10/2004 on more harmonised information provisions. 11987/04/EN, WP 100, [http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2004/wp100\\_en.pdf](http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2004/wp100_en.pdf), November 2004.
- [Pea06] S. Pearson. Towards automated evaluation of trust constraints, in trust management. In *Trust Management, LNCS 3986*, pages 252–266. Springer Berlin/Heidelberg, 2006.
- [Pet05] J.S. Pettersson. HCI guidance and proposals. PRIME Deliverable D6.1.c, February 2005. [https://www.{PRIME}\-project.eu/{PRIME}\\\_products/reports/arch/](https://www.{PRIME}\-project.eu/{PRIME}\_products/reports/arch/).
- [PFHD<sup>+</sup>05] J.S. Pettersson, S. Fischer-Hübner, N. Danielsson, J. Nilsson, M. Bergmann, S. Clauss, T. Kriegelstein, and H. Krasemann. Making PRIME Usable. In *SOUPS 2005 Symposium on Usable Privacy and Security, Carnegie Mellon University*, 2005.
- [PKHvB03] A.S. Patrick, S. Kenny, C. Holmes, and M. van Breukelen. Human computer interaction. In J.J. Borking & J.G.E. Olk G.W. van Blarckom, editor, *Handbook of Privacy and Privacy-Enhancing Technologies: The Case of Intelligent Software Agents*, pages 249–290. College Bescherming Persoonsgegevens, Den Haag, The Netherlands, 2003.
- [Pri08] PrimeLife WP4.3. UI Prototypes: Policy administration and presentation – Version 1. In Harald Zwingelberg Simone Fischer-Hübner, editor, *PrimeLife Deliverable D4.3.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, June 2008.
- [Pri09a] PrimeLife WP4.1. Low-level Prototypes. In Christina Köffel and Peter Wolkerstorfer, editors, *PrimeLife Deliverable D4.1.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2009.
- [Pri09b] PrimeLife WP4.2. Trust and Assurance Control – UI Prototypes. In Jenny Nilsson Simone Fischer-Hübner, editor, *PrimeLife Deliverable D4.2.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, June 2009.
- [Pri09c] PrimeLife WP4.3. UI Prototypes: Policy Administration and Presentation – Version 1. In Simone Fischer-Hübner, Erik Wästlund, and Harald Zwingelberg, editors, *PrimeLife Deliverable D4.3.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, June 2009.
- [Pri10a] PrimeLife WP4.1. HCI Pattern Collection – Version 2. In Simone Fischer-Hübner, Christina Köffel, Erik Pettersson, John-Sören Wästlund, and Harald Zwingelberg, editors, *PrimeLife Deliverable D4.1.3*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, February 2010.
- [Pri10b] PrimeLife WP4.1. High-level Prototypes. In Cornelia Graf, Peter Wolkerstorfer, Erik Wästlund, Peter Wolkerstorfer, Simone Fischer-Hübner, and Benjamin Kellermann, editors, *PrimeLife Deliverable D4.1.4*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, August 2010.
- [Pri10c] PrimeLife WP4.2. End User Transparency Tools: UI Prototypes. In Erik Wästlund and Simone Fischer-Hübner, editors, *PrimeLife Deliverable D4.2.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, June 2010.

- [Pri10d] PrimeLife WP4.3. UI Prototypes: Policy Administration and Presentation – Version 2. In S. Fischer-Hübner and H. Zwingelberg, editors, *PrimeLife Deliverable D4.3.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, June 2010.
- [Pri11] PrimeLife WP4.1. Final HCI Report. In Simone Fischer-Hübner et al., editor, *PrimeLife Deliverable D4.1.5*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2011.
- [Pro10] Privicons Project. Privicons project. <http://www.privicons.org/projects/icons>, 2010.
- [PWG10] Stefanie Pötzsch, Peter Wolkerstorfer, and Cornelia Graf. Privacy-awareness information for web forums: results from an empirical study. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 363–372, New York, NY, USA, 2010. ACM.
- [Ras10] A. Raskin. Privacy icons Making your online privacy rights understandable. <http://www.drumbear.org/project/privacy/-icons>, 2010.
- [Rod03] W. Rodger. Privacy isn't public knowledge: Online policies spread confusion with legal jargon. *USA Today* (May 1, 2003, 3D), 2003. Last Visited: 28.04.2010.
- [RSM05] J. Riegelsberger, M.A. Sasse, and J. D. McCarthy. The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies*, 62(3):381–422, 2005.
- [Run06] M. Rundle. International data protection and digital identity management tools. Presentation at IGF 2006, Privacy Workshop I, Athens, 2006.
- [SBHK06] Steve Sheng, Levi Broderick, Jeremy Hyland, and Colleen Alison Koranda. Why johnny still cant encrypt: Evaluating the usability of email encryption software. In *SOUPS: Proceedings of Symposium On Usable Privacy and Security*, 2006.
- [SF05] M. A. Sasse and I. Flechais. Usable security: What is it? how do we get it? In Lorrie Faith Cranor and Simson Garfinkel, editors, *Security and Usability: Designing Secure Systems that People can Use*. O'Reilly Books, 2005.
- [SGB01] Sarah Spiekermann, Jens Grossklags, and Bettina Berendt. E-privacy in 2nd generation e-commerce: privacy preferences versus actual behavior. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 38–47, New York, NY, USA, 2001. ACM.
- [SSM10] R.I. Singh, M. Sumeeth, and J. Miller. *A User-Centric Evaluation of Readability of Privacy Policies in Popular Web Sites*. Springer, 2010.
- [TERS<sup>+</sup>06] Janice Tsai, Serge Egelman, Kok-Chie Daniel Pu Rachel Shipman, Lorry Cranor, and Alessandro Acquisti. Symbols of privacy. In *Poster Proceedings of the Symposium of Usable Privacy and Security (SOUPS 2006)*, July 14-16 2006.
- [TS04] S. Tullis and J. Stetson. A comparison of questionnaires for assessing website usability. In *Usability Professional Association Conference*, 2004.
- [TZY01] C. W. Turner, M. Zavod, and W. Yurcik. Factors that affect the perception of security and privacy of e-commerce web sites. In *Proceedings of the Fourth International Conference on Electronic Commerce Research*, Dallas TX, November 2001.
- [WP008] WP06.1. HCI Guidelines. In John Sören Pettersson, editor, *PRIME deliverable D6.1.f*. PRIME Project, February 2008.
- [WT99] Alma Whitten and J. D. Tygar. Why Johnny can't encrypt: a Usability Evaluation of PGP 5.0. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 14–14, Berkeley, CA, USA, 1999. USENIX Association.
- [WTS<sup>+</sup>08] Peter Wolkerstorfer, Manfred Tscheligi, Reinhard Sefelin, Harald Milchrahm, Zahid Hussain, Martin Lechner, and Sara Shahzad. Probing an agile usability process. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 2151–2158, New York, NY, USA, 2008. ACM.
- [WWK10] E. Wästlund, P. Wolkerstorfer, and C. Köffel. PET-USES: Privacy-Enhancing Technology - Users' Self-estimation Scale. In M. Bezzi, P. Duquenoy, S. Fischer-Hübner, M. Hansen, and G. Zhang, editors, *Privacy and Identity Management for Life*, pages 266–274. Springer Boston, 2010.

## **Part IV**

# **Policy Languages**

## Introduction

Machine-interpretable policy languages are at the heart of any modern privacy infrastructure. Rather than “hard-coding” fixed privacy policies into the infrastructure, dedicated policy languages provide the flexibility to express and change policies without having to re-implement the software that enforces them. Moreover, if multiple interacting parties agree on the grammar and semantics of a language, policy languages can also be used to communicate privacy policies across different interacting entities. Finally, security and privacy policy languages are an important tool to ensure compliance with legal, industrial, and user requirements.

PrimeLife set out to collect policy language requirements from the diverse scenarios covered by the project, and to analyze the suitability of existing policy languages to cover the privacy aspects. It quickly became clear that none of the existing languages covered all the needs. A report of these activities is given in Chapter 16. However, it also became clear that satisfying all of the collected requirements was far beyond PrimeLife’s available time and budget. We therefore hand-picked a number of features based on their potential to improve digital privacy in the real world and on their feasibility within the restrictions of the PrimeLife project.

Chapters 17 and 18 give an overview of the technical research results on two main aspects in which appropriate policy language support was found lacking. Chapter 17 focuses on the relation between access control policies, which specify which entities are allowed to obtain certain information, and data handling policies, which specify how these entities are to treat the obtained information. This relation becomes particularly complex if information can be forwarded to third parties, so-called downstream usage. The chapter describes a language that allows users to automatically match their preferences to the policies proposed by servers, thereby assisting them in their choice of whether or not to reveal their information.

Chapter 18 focuses on privacy-friendly access control policies. It proposes “credentials” as a generalisation of several existing authentication technologies, covering well-established technologies such as X.509 certificates and LDAP directories, as well as anonymous credentials. Rather than assuming that all of a user’s attributes are revealed by default, the language expresses which credentials users need to possess in order to gain access, which attributes they have to reveal, and which conditions they have to satisfy. Policy sanitization strikes a balance between users’ privacy and enterprises’ sensitivities about the policy details.

Chapter 19 takes a closer look at the legal requirements under European law to transparently inform users about how their information is used. Expressing such usage in an understandable way is a notorious challenge. Faced with the multitude of applications and usage purposes and the lack of a structured ontology among them, this chapter investigates the current practices in data usage in various contexts and discovers a common structure.

Finally, a number of these concepts, in particular the research results presented in Chapters 17 and 18, were brought together in the design of the PrimeLife Policy Language (PPL). To be of use in real-world settings, PPL is defined as extensions to the industrial standards XACML and SAML. Chapter 20 reports on the architec-

ture and implementation of a fully functional policy engine to enforce PPL, thereby bringing the advanced research concepts to life.





# Chapter 16

## Policy Requirements and State of the Art

Carine Bournez and Claudio A. Ardagna

**Abstract** The design and implementation of a versatile privacy policy language is one of the core activities in the PrimeLife project. Policy languages are a crucial tool in any privacy-aware information infrastructure. Machine-interpretable languages have a major advantage over natural languages in that, if designed properly, they allow automated negotiation, reasoning, composition, and enforcement of policies. The requirements are the first step in the development of such a language. The methodology was to collect use case scenarios and derive concrete requirements from them. This chapter presents those requirements independently; they are not derived from research work other than the PrimeLife study itself.

### 16.1 Definitions

We first define the three types of policies that, in our view, are important parts of any privacy policy: *data handling*, *access control*, and *trust policies*. This by no means implies that we consider these to be separate, independent policies that together form the privacy policy. Rather, we see them as three minimal aspects that have to be covered by any policy language. There may be other aspects, and the three aspects mentioned here may not be orthogonal.

#### 16.1.1 Data Handling Policies

A *data handling policy* (DHP) is a set of rules stating how a piece of sensitive data should be treated. In the context of privacy, we are mostly interested in the case where a piece of data is personally identifiable information (PII). The data handling policy specifies, among other things, for what purposes the data can be used (e.g., research, marketing), to which third parties the data can be disclosed

(e.g., all, nobody, only auditors), and the obligations on data management (e.g., how long the data can be stored). Obligations define actions that must be executed by the party in charge of enforcing a policy. Those actions are triggered by events such as time or handling collected data.

We distinguish between three different types of DHPs, namely *data handling preferences* on the data subject's side, and *data handling policies* and *sticky policies* on the data controller's side. In the *data handling preferences* associated with a piece of PII, the data subject specifies its requirements on how it expects the data to be treated by the data controller. The data controller, before receiving the PII, describes his intentions on how he will treat the PII in his *data handling policy*. When an agreement is reached, the agreed-upon policy that the data controller is required to adhere to is referred to as the *sticky policy*.

### 16.1.2 Access Control Policies

An *access control policy* (ACP) protects access to an object by specifying which recipients should be granted which type of access to the object. The object being protected can be a piece of data such as a file, a database record, or a webpage, but it can also be a more abstract functionality such as a service or a remote procedure call. The subject can be specified by means of a unique identifier (e.g., user name), by role (e.g., administrator), by a group that he belongs to (e.g., helpdesk), or by other attributes (e.g., age, reputation, ...). A subject can be any type of entity that is capable of making a request; it could be a natural person but could also be a running process or a device, or a combination of these. The possible types of access (e.g., read, write) depend on the resource that is being protected. Finally, the decision to allow or deny access can be based on the subject's properties, the content of the resource, the details of the access request (e.g., parameter values passed in a remote procedure call), and secondary information such as current time, processor load, etc.

### 16.1.3 Trust Policies

The concept of *trust* is almost inherently vague due to its close association with the subjective decisions made by humans in real life. Even within the technical community, there seems to be quite some confusion about the definition of trust policies. In the scope of this chapter, we clarify the concepts related to trust that would be required in a policy language. The list of requirements will also contribute to delimiting the technical definition of trust.

In general, a trust rule is a rule expressing that a specific entity is entrusted to perform a specific action if a specific condition holds. When placing trust in someone or in an entity, the relying party expresses a belief that that person or entity will behave in a way that is beneficial to the relying party's interests, and will not behave

in a way that would harm its interests. Trust is contextualised, in that your trust in someone to perform certain actions is limited to within a given context. The action can be either to certify a specific piece of content (e.g., write about technology or issue a passport) or to adhere to an agreed-upon data handling policy. The condition can be any condition that has to be satisfied for the rule to apply. We leave open what kind of statements the condition can contain; conditions can be based on the credentials held by X, privacy labels, reputation, environmental conditions, etc.

The two actions that we have in mind when talking about trust are trust in an entity to certify a specific type of content (*content trust*) and trust in an entity to adhere to an agreed-upon data handling policy (*data handling trust*). The main difference between both types is the direction of the information flow. For content trust, it is the trusting entity who receives information (possibly indirectly) from the trusted entity; for data handling trust, it is the trusted entity who receives information from the trusting entity. Content trust is about the correctness of received information; data handling trust is about how information is treated after it is transmitted.

## 16.2 Legal Requirements

Besides use cases and applications where policy languages will be used for privacy protection, legal requirements are imposed by institutions (governments, European Union). We briefly summarise the main legal requirements, that will later be derived into technical constraints on the policy language.

The processing (including collection, storage, retrieval, transferral, and other means of handling) of any data that can be linked to a person (personal data, for a more thorough definition see Article 29 WP, Op. 136) by another entity (a data processor) has to be legitimate. If processing takes place without obeying legitimacy, those subjected to the processing (data subjects) would lose trust in markets and tend to not give away their data when acting in these markets. Therefore, data protection is a market enabler, but above all it is recognised as a fundamental right, and is acknowledged by many constitutions in the European Union, as well as in the Charter of Fundamental Rights of the European Union (cf. Art. 8 thereof) and the European Convention on Human Rights (Art. 8 thereof).

On an operational level, the Directive 95/46/EC on the protection of personal data (Data Protection Directive) [Dir95], and the Directive 2002/58/EC on Privacy and Electronic Communications (E-Privacy Directive) provide the baseline for compliance. In many areas, sector specific regulation and national implementation of directives need to be taken into account. Both directives, as well as most of the other regulations, follow a set of well established principles, with the principle of fair and lawful processing, the purpose limitation principle, data minimisation, and the transparency principle at their core to name a few.

**Fair and lawful processing.** Conceptually, European law effectively prohibits any processing except where there is a legal basis (Art. 6 of 95/46/EC). This means that in a professional context, handling data without a legal basis is illegal. Non-

compliance can result in penalties and may even lead to data protection authorities shutting down IT systems (see §38 of the German “Bundesdatenschutzgesetz”). A legal basis can be derived directly from legal regulation, e.g., when the law prescribes the storage of specific data for law enforcement purposes. In many cases, however, the lawfulness can be achieved by receiving a consent, in an unambiguous form, from the person whose data is concerned, i.e, from the data subject (Art. 7(a) of 95/46/EC).

**Purpose limitation.** The processing of personal data – even if acquired lawfully (which should result in legitimacy) – is still a subject to further regulation. It regularly needs to follow, amongst others, the principle of purpose limitation (Art. 6.1(b) of 95/46/EC). Put simply, the purpose limitation principle states that data may only be collected, stored, processed or transferred for those purposes of which the data subject has given consent, or of which the law allows. No further processing that would be incompatible with the original purpose is allowed.

**Data minimisation.** This implies that if no purpose is at hand, the data has to be deleted or not even collected in the first place (Art. 6.1(c) 95/46/EC). But data minimisation can be understood in a broader sense to construct systems in such a way that processing personal data can be avoided (so-called data avoidance). While the former is a legal requirement, the latter is not mandatory by law in all cases, but only where such technology is available under reasonable conditions (cf. Recital 46 of 95/46/EC). However, it can be sensible to develop and use such approaches even in an enterprise context when there is no legal necessity, as it may lower compliance costs.

**Transparency and subject access rights.** The principles regarding the processing itself are adjunct to specific, enforceable rights for the data subject (e.g., Art. 12 and 14 of 95/46/EC). The conceptual idea behind these rights is that the data subject should be able to find out what others know about him or her. In case this knowledge is illegitimate, the data subject should be able to stop the respective data controllers from using this knowledge, by blocking, correcting or deleting the personal data.

Generally, the protection of data does not always prove to be easy in information systems. This is especially true for the protection of personal data. A core difficulty lies in the diversity of the processing steps that this data may undergo, while at the same time being subjected to the above principles. For compliance, it has to be ensured that any algorithm or any service of an IT system that processes a specific set or piece of personal data is within the limits of the legal foundation (e.g., the consent) for the processing, and that it does not violate the purpose limitation principle. At the same time, it needs to be ensured that the data subject is able to find out what happened to his or her data, who accessed it, and what it has been or will be used for.

## 16.3 Policy Language Requirements

### 16.3.1 General Design Principles and Expressivity

**Measurability.** This property of a policy language is fulfilled when the construction of the language allows one to check that the policy has been followed correctly. A mechanism to prove that a rule has been applied is useful but not sufficient to demonstrate this property, since some policies can be applied a long time after the moment when it has been stated.

**Unified model.** Because Data Handling Policies are closely related to Access Control policies, a unified model is a key success factor for a policy language. Even though in this document we often focus on access control and data handling policies separately, they are in fact closely related. A server's access control policy should not only specify what PII it wants from the user, but also how it is planning to treat the data. Here, the DHP is part of the ACP. On the other hand, a user's DHP may specify which third parties are allowed to see the PII, so that the ACP becomes part of the DHP.

**Semantic compatibility with P3P.** A policy language should be semantically compatible with the Platform for Privacy Preferences (P3P [W3C06a]) as much as is possible.

**Stickable policies.** *Stickability* is the property of the policy language that allows for attaching a policy to data no matter how, where and when the data is sent.

**Revocability.** It must be possible for any policy user to revoke a Data Handling Policy the same way it is possible to revoke a credential related to an Access Control Policy.

**Transparency.** A policy language must be able to express that the data flow trace resulting from the transfer of the data between entities should be kept. There should be mechanisms in place to log the usage of personal data. Such logs will span multiple trust domains in case of downstream usage and parts of it may travel with the data (sticky logs).

**High-level (abstract) policies.** The policies should be expressible not only on a low, i.e., more technical, level but also on a higher, i.e., more abstract, level. The benefits of this are, for example, that the policies can become shorter, easier to understand and easier to formulate. Among other techniques, ontologies could be leveraged to bring the policies to a high(er) level. Instead of talking about `credit-card-number`, `credit-card-name`, etc., an ontology could describe such data under the class `credit-card data`, or even more general, `payment data`. Then the policy can refer to concepts like `credit-card` or `payment data` if needed.

**Data minimisation.** The policy language should support – and encourage – the minimisation of the amount of personal information that is revealed in order to gain access to a resource. The architecture should definitely not assume that all informa-

tion about the subject is readily available when the access decision is made. Rather, the list of attributes that need to be revealed, or the predicate that needs to be proved, should be explicitly specified by the server, or perhaps even be the result of a negotiation between the client and the server. The client should then have the option to reveal only those attributes that are strictly necessary. Whether this is possible, of course, not only depends on the policy language, but also on the underlying authentication technology.

**Anonymous or pseudonymous access control.** A user shall have the possibility to access a resource in an anonymous or pseudonymous way. For an anonymous access, the server makes sure that the user fulfills the necessary requirements, while the required attributes allow the user to stay anonymous. This is of course only possible if (1) the required attributes are applicable to a large number of people and the user can therefore not be identified, and (2) the underlying technology supports proving the attributes in an anonymous way (for example using the technology of *anonymous credentials*). A pseudonymous access is similar to the anonymous one, with the difference that for every access a user makes, he provides some kind of identifier - a pseudonym - which the server uses to recognise that the same (pseudonymous) user requests access. However, the server only knows the pseudonym and not the real identity of the user. This is important if a user wants to keep some profile on the server side, while the user still wants to be anonymous to the server. From a legal point of view, services must be offered pseudonymously whenever that can be considered reasonable for the service.

**Meta-policies and policy generation.** In some cases, it is necessary to constrain the possible policies that can be attached to data by rules or guidelines. These guidelines are locally enforced when defining preferences and policies. For instance, a data subject may define a rule (i.e., a policy) that forbids the creation of any preference allowing the use of medical data for advertisement. Such a policy could also be provided by a trusted third party. This can be achieved by defining meta-policies that specify how policies can be customised. The same mechanism can be used to specify constraints on policies that are generated, e.g., by a service in response to a user request. Those constraints can be derived from trust or access control assertions. This mechanism can rely on a way to express policy generation rules in the policy language itself.

**Data model primitives.** The policy language must make consistent use of (at least) the concepts of date and time, and location. These concepts are essential for the expression of data usage constraints, e.g., some data can be displayed in some particular locations or can be valid for a limited period of time.

### *16.3.2 Requirements for Data Handling Policies*

**Business logic to describe data usage.** The business logic of an enterprise determines what actually happens with the data after it is received. If this business logic

is described in a standardised way, for example using WS-BPEL (Business Process Execution Language), then it should be possible to automatically derive the DHP from it, or perhaps the BPEL description itself could even be part of the DHP.

**New usage should trigger consent.** The policy language must support a mechanism to acquire new consent from the data subject if the data controller wants to change the policy. However, the data subject can indicate in her preferences that she will never agree to any changes to the policy, and that she therefore does not want to be bothered with requests for policy changes. Alternatively, one could have an opt-in mechanism, where the data subject has to explicitly state in her preferences that she would consider changes to the policy.

**Legal policies need differentiated layering.** A policy language must include the possibility to express and address at least three layers of human-readable text to describe a policy to the user. This is recommended by the Op. 100 of the Article 29 group's Opinion on More Harmonised Information Provisions: a short version of the privacy policy, with an addressable substructure to be defined; a condensed version of the privacy policy, with an addressable substructure to be defined; and a full (lawyers readable) version of the policy, with an addressable substructure to be defined. A fourth layer to express the policy with iconography should also be available, with a set of icons to be defined.

**Technical representation of legal policies.** The policy language should be able to express legal policy concepts (e.g., liability, data controller, data processor, etc.) and conditions relevant for machine-based decision making, in a form supporting their digital storage, transmission, and processing. The semantics of the representation should be carefully considered and be compatible with the capabilities of an efficient processing engine.

**Constrained customisation of privacy policies.** In specific cases, it is necessary to let the data subject create a sticky policy that is slightly different from the privacy policy of the data controller. We assume that 1) the data controller constrains which modifications are legitimate and 2) the data controller checks whether the provided sticky policy is indeed compliant with the initial policy.

**Support nested policies.** The policy language must support nested policies. Thus, a policy could include a number of specific policies for further processing of the data.

**Express user preferences.** A policy language must allow for users to express preferences about the handling of their data. In particular, it must be possible to express preferences for the use of given credentials for a given purpose. The user should also be able to express general trust relationships independently of a given scenario or purpose. The language should be extensible enough to express new user-defined preferences.

**Describe server policies.** When the server has its own Data Handling Policies (one or multiple), the user's Data Handling Policy should match one of the available server policies.



**Originator's policy.** A policy language must include a mechanism to identify what the data subject allows, no matter who transmits the data.

**Logging/Monitoring/Auditing Policies.** It must be possible to inform the user about data collected during the usage of the service (date, location, actions, credentials used, etc.). It must be possible to express the scope of the retention (page, session, duration) and usage (extend user experience, debugging, legal requirements) of the collected data. It must be possible to express how the data is collected: how, when, where it is stored.

**Security levels.** The level of privacy protection achieved by setting a policy does not only depend on the claims made by the subject, but also on the underlying technology that is used to prove the validity of these claims. The policy designer should not be concerned with technical details such as cryptographic algorithms and key length, but given that the language should be useful in both low- and high-security environments, some notion of "security levels" seems appropriate. What these security levels imply on the underlying technology and infrastructure could then be specified in a separate ontology. The policy designer could use this ontology in a more practical way than defining the technical details himself.

**DHP ontology.** The language should provide an ontology for data purposes and types of data. It should be extensible, as we can impossibly foresee all items that should appear in this ontology.

**Enforcing DHP.** Technological means to enforce Data Handling Policies are limited. A trusted software infrastructure can assist in automatically adhering to a DHP (e.g., deleting data on time) and in logging access for audit purposes, but eventually these systems can always be circumvented by a malicious user (e.g., by forwarding a picture of the screen displaying the sensitive information). In the end, one will have to either trust the receiver to adhere to the DHP that was agreed upon, or to trust an external auditing agency to correctly certify such receivers. It should be possible to express this type of trust in the policy.

**Breaking the glass.** A special case of a policy: the law prescribes certain areas, where access and processing is compliant, although clearly not within the prior consent. In these cases, it might be reasonable to invoke special mechanisms for transparency (i.e., the glass is broken = the prior consent has been exceeded). Such a policy could state that certain entities are entitled to access the data, but then certain obligations regarding information of other parties, particularly the data subject or data protection supervisors, might be triggered.

**Capture user intent.** This property is needed to differentiate the user's intent when using a service from the purpose of the service itself. User intent does not necessarily match the service purpose. The mechanism for capturing the user intent may be simple, however, processing it requires use of semantics and logic.

**Purpose of data processing.** The purposes of data processing or data handling by a service usually stay the same across the usage of the service for all transactions. However, it is not always clear for the user whether a given piece of data is going to

be reused for a purpose other than the most obvious one. For example, an address may be used for shipping, but also for later marketing actions. The policy language must allow for expressing all the different purposes of data handling by the service, so that the user can be informed about the less obvious purpose(s) of data processing.

**Express obligations.** A policy language needs to express all the obligations of the processing party. Such obligations can be derived from the law, but also from consent. Full coverage by a machine readable policy will never be achievable, as the law often requires human interpretation (that is why we have judges). Obligations should be ready to cover the scope of purpose limitations, which is difficult to translate, since it is hard to describe, whether an access, storage, or transfer was made for one or another purpose.

**Notification/feedback channels.** It should be possible for the data subject to require that notification messages be sent back to him to inform him about the obligation enforcement conditions of his exported data. This is an important feature since few security systems are able to provide a report to the data subject about the usage of his private information. In case of misbehaviour, these notification messages can be used as a proof for accountability. It should be possible to link different notification messages for the same piece of exported data together, so that they form a trace of the data usage and possibly a proof of privacy policy violation.

### *16.3.3 Requirements for Access Control policies*

**Declarative language to represent access control policies.** The policy model should be accompanied by a language that enables the specification of access control policies. The language should be declarative and accompanied by a clear and unambiguous semantics for the policy specifications.

**References to policies.** The policy model should provide support for reusing a policy. Referencing can be done either directly or by inheritance.

**Role models - family, friends, wider access control.** The access control model should support multiple access control paradigms, including role-based access control and attribute-based conditions. Roles could also be incorporated in attribute-based conditions by the consideration of proper attributes.

**Information from third party sources.** The policy model/language should be able to leverage information certified by a given third party (e.g., government).

**Technology-independent credentials.** The policy model should support expressions on attributes contained in digital credentials. Different types of credentials may be integrated in the policy model/language, such as anonymous credentials, X.509 credentials, pseudonym/password, Kerberos tickets, etc.

**Attribute-based access control to data.** The policy model/language should support policies making explicit references to attributes of involved parties (e.g., requester of access, data on which access is requested, respondent/owner of data). Attribute values can be provided by means of credentials or can be metadata associated with objects.

**Expiration date, validity.** There should be an option for access control policies to expire after an amount of time. Access control policies should support conditions and reasoning about time. Time can impact the validity of certain conditions in the policies or be used to support policies that might be valid only up to, or after, a specific time (e.g., embargo on data, data that become public after a given time, data that should be deleted after a given time).

**Time or event for the beginning of validity.** Access may be granted or denied for a user or groups of users after an amount of time or after an event occurred (for instance to support history scientists etc.). The policy model may support event-based conditions other than those expressed as a time. Event-based conditions make policy restrictions/permissions valid at the occurrence of certain events.

**Priority of policies or combination rules for policies.** In case of contradicting policies, we need a clear prioritisation; that is, a rule that determines which policy supersedes all others and how the others are combined with that policy and with each other (you may think of a hierarchy of policies as well). The policy model should support a mechanism for combining policies according to different composition operators. The policy model/language should be accompanied by a clear definition of the possible composition operators and their semantics should be provided.

**Data subject should be able to keep control over PII.** A data subject should be able to control (modify, delete, etc.) his PII that has been collected and stored by a data controller. In this case, the Access Control of the data controller has to take into account the obligation to let the data subject access his PII. The control of the data subject on his PII may range from getting read access to stored PII, being allowed to update collected PII, accessing logs regarding the usage of collected PII, to modifying sticky policy referring to collected PII, etc. For instance, a data subject provides his home address (PII) to a data collector. Subsequently, the data subject may use a dedicated endpoint (e.g., WS, Web page) offered by the data collector to access and check logs related to the usage of his PII in order to figure out whether his PII was used appropriately.

**Choose strength of protection.** The policy model should provide different levels of protection and give users the ability to tune protection according to their needs. For instance, requesting the application of specific cryptographic measure in communication or storage of private information.

**Ontologies for credential types, delegation.** Ontologies can be a powerful tool to adapt the language to the particular needs of a particular context while maintaining interoperability. For example, a common ontology on the structure of personal identity information can be used to guarantee compatibility of digital passports issued

by different countries. An ontology on countries and their governments can be used to determine which instances are certified to issue passports for which countries.

### ***16.3.4 Requirements for Trust policies***

**Link to Data Handling policies.** The policy language should provide the possibility to link a Data Handling Policy with trust policies; this would permit the explicit representation of trust on the correct enforcement of the Data Handling Policy. When the binding with the trust policy is not expressed, trust is implicitly assumed (this is often the case when trust is established at a more global level). The binding between Data Handling and Trust Policy could occur at different levels, expressing requirements on the data subject or on the source of the credentials referred to in the policy.

**Trust establishment.** Several factors should be taken into account in the decision to trust another entity or not. A first factor could be the exchange of credentials. Trust could also be based on statements made by others, for example reputations or privacy seals.

**Statement and Certification.** Certification validates that a server is authentic and trustworthy, so that the user can feel confident that their interaction with the server has not been overheard and that the server is who it claims to be. The certificate is provided by a third party that should be trusted by the user.

- A trust statement is the explicit expression of a perceived trust level. It is made by the truster and represents the subjective judgement of the trustee's trustworthiness, according to the truster's point of view.
- A certificate is a digital document that describes a written statement from the issuer (certification authority, often considered trustworthy) about the identity of another party (in the form of public key and the identity of the owner) or a permission to use a specific service. It can be considered as a trust statement issued by a reputed third party.

**Context-dependent Trust Mechanisms.** Trust mechanisms should be chosen according to the application context. For example, a trust policy should express a rule specifying that the data subject provides her e-mail address to an online book store, if its reputation is greater than 8/10, and to an online tax declaration website, if it is certified by the government.

**Privacy breach.** Privacy breaches include (but are not limited to) loss of control on data, unauthorised access control, social engineering, phishing, and malicious proxy server. In some cases, legislation requires that when a privacy breach occurs, a notification has to be sent to the affected individual or organisation. Technological means to enforce these rules have to be put in place, as well as a formal and quantitative definition of *privacy breach* assessment to evaluate when the breach occurs and what is the level of associated risk.

**Link trust with Access Control.** Conditions on policy may include trust evaluation, e.g., allow data writing if the user complies with ACP and has a trust level greater than  $X$  or if he has a certain certification.

**End user trust.** Trust affects all levels of end-user interaction with the system, i.e., whenever a user wants to access a service on the web. Trust should be assessed for all layers involved in the transaction: user application, network, service provider. Trust is by definition related to a personal perception, so each user has to be able to edit trust policies and trust preferences in an intuitive way.

**Building trust through a third party.** Users may establish trust relationships using third party trust assessment. This may guarantee a maximum level of trust equal to the level of the certification authorities (best case scenario). Trust mechanisms have to support certificates as produced by certificate authorities (e.g., CAcert, Thawte, etc.) and the corresponding hierarchical mechanisms (web of trust). Trust reasoning has to allow for combining this information with other trust metrics (e.g., reputation based).

**Trust reasoning.** A trust policy's evaluation component should be able to reason about trust, including composing various trust metrics (e.g., reputation system, PKI ...) and hierarchical structures.

**Trust ontologies.** Trust credentials and trust assessment mechanisms should be represented in an ontology. This ontology should categorise the different types and sub-types of credentials. For each type of credential we can attach an information about the trust assessment mechanism supporting it.

**Transparency, reciprocity.** Transparency should be considered as one component of trust assessment (typically, transparency increases trust perception). If a data holder is able to monitor at any time the usage of his data by a server, his trust feeling will increase. Due to the fact that transparency techniques may include: historical data, previous behaviours, access to log files etc., this requirement is strongly related to the requirement on Logging/Monitoring/Auditing Policies. Reciprocity should be taken into account as characterised trust interaction, but this is not the general case, e.g., I trust a mail provider for storing my personal mails but it does not necessarily trust me for storing the same kind of information.

### **Specification of liabilities.**

- Towards data subject: data protection obligations under the 95/46 Directive have to be fulfilled by data controllers. Data controllers are liable for data protection violations unless they can prove they are not responsible for the damage. It is necessary to differentiate between data controller and data processor. The role of data processor is reduced: he solely processes personal data as directed by the controller. The policy language should be able to express the role of each entity for each action to determine who is liable. Liability: compensation from the controller for the damage suffered. Remedy: a right to a judicial remedy for any breach of guaranteed rights. Sanctions: to be defined by member states in their national laws.

- Towards relaying parties: for ensuring data accuracy, the following policies are important: validation of data at the moment of collection, procedures for reporting and dealing with suspected inaccuracies, regular updates, restriction of modification rights to authorised entities.

### ***16.3.5 Other Technical Requirements for PrimeLife***

The following requirements are specific to composition scenarios and use cases where anonymous credentials are used. They are lower level requirements that influenced technical design and some can also be viewed as technical choices.

#### **Policy Composition.**

- Support for composition of service policies and composition of user preferences.
- Cascading policies: When rules are defined at different levels (e.g., corporate, service, and action), mechanisms to select and aggregate appropriated rules should be provided.
- Prioritisation of rules: Priorities are only necessary to resolve conflicts between rules. Depending on the expressiveness of the language, priorities may be required.
- Generalisation of policies.
- Multi-rounds policy definition.
- Policy negotiation: Negotiation only makes sense when the user and/or the service have a trade-off to make.
- Delegation of Rights.
- Revocation of Rights.
- Composition of Access Control Policies.
- Prior agreement and contracts.
- Privacy-aware audit mechanism.
- Support for data and PII: Legislations treat personal data (PII) differently from other types of data.
- Dynamic Trust: mechanisms to bootstrap, modify, and revoke trust are necessary.
- Scope: Trust is not unconditional. The scope of the trust relationship has to be defined.
- Proof of enforcement.

#### **Use of Anonymous Credentials.**

- Technology-independent certification of data by trusted third parties.
- Trust in certified data.
- Predicates over attributes, extensible with ontologies.
- Expression of proved statement (by using same policy language).
- DHP of Derived PII: sensitive information that is computed based on actual PII.
- Alternative data recipient + associated access conditions.
- Notion of atomic credentials.

- Limited spending: The policy language should have provisions to express that one can only authenticate oneself with the same credential for a limited number of times.
- Signing statements.

## 16.4 State of the Art

Technical improvements of Web technologies have fostered the development of on-line applications that use private information of users to offer enhanced services. As a consequence, the vast amount of personal information thus available on the Web has led to growing concerns about the privacy of its users that require the ability to communicate in a secure global networked infrastructure while at the same time preserving their privacy. Support for digital identities and credentials, and definitions of access control and privacy-enhanced languages, protocols, and techniques for their management and exchange then become fundamental requirements.

### 16.4.1 Access Control Policy Languages

Several access control models and languages presented in the literature [DFJS07, SD01] are based on logic expressions, and prescribe access decisions on the basis of some properties that the requesting party may have. These properties can be proven by presenting one or more credentials [BS02, IY05, LMW05, NLW05, YWS03]. Credential-based access control can be seen as a generalisation of a variety of access control models. In (hierarchical) role-based access control (RBAC) [FK92, SCFY96], the decision to grant or deny access to a user is based on the roles that were assigned to her. Clearly, one could encode the roles of a user in a credential, so that RBAC becomes a special case of credential-based access control. However, RBAC is not powerful enough to support the concept of credential.

Attribute-based access control (ABAC) [BDDS01, eXt05, WWJ04] comes closer to the concept of credential-based access control, since it grants access based on the attributes of a user. The de facto ABAC standard *eXtensible Access Control Markup Language* (XACML) [eXt05] is an OASIS standard that proposes an XML-based language for specifying and exchanging access control policies over the Web (see also Sections 18.4 and 19.3.1). The language can support the most common security policy representation mechanisms and has already found significant support by many players. Moreover, it includes standard extension points for the definition of new functions, data types, and policy combination methods, which provide a great potential for the management of access control requirements in future environments. Though XACML represents the most accepted, complete, and flexible solution in terms of access control languages, it only allows the specification of the issuer of the attributes, but does not see them as grouped together in atomic creden-



tials. Moreover, the architecture paradigm is far from privacy-friendly: the user is assumed to provide the policy decision point (PDP) with all her attributes, and lets the PDP decide on the basis of its access control policy. The policy that needs to be satisfied is not known to the user, leaving no opportunity for data minimisation.

The first proposals that investigate the application of credential-based access control regulating access to a server are done by Winslett et al. [SWW97, WCJS97]. Access control rules are expressed in a logic language, and rules applicable to a service access can be communicated by the server to the clients. A first attempt to provide a uniform framework for attribute-based and credential-based access control specification and enforcement is presented by Bonatti and Samarati [BS02]. The authors propose a language for specifying service access and information release rules based on credentials with certain properties. Access rules are specified as logical rules, with some predicates explicitly identified. Attribute certificates are modeled as credential expressions. In addition, this proposal also permits reasoning about declarations (i.e., unsigned statements) and profiles of the users that a server can make use of to reach an access decision.

Besides solutions for uniform frameworks supporting credential-based access control policies, different automated trust negotiation proposals have been developed [LWBW08, SWY01, YW03]. Trust is established gradually by disclosing credentials and requests for credentials. The work in [WSJ00] describes how trust can be established through the exchange of credentials. The authors present a credential-based access control language that is used for protecting the user credentials. The work by Ni et al. [NLW05] takes the idea of [WSJ00] to cryptographic credentials and defines a grammar for a revised version of the policy language. Trust management systems (e.g., Keynote [BFIK98], PolicyMaker [BFL96], REFEREE [CFL<sup>+</sup>97], and DL [LGF00]) use credentials to describe specific delegations of trust among keys and to bind public keys to authorisations. They therefore depart from the traditional separation between authentication and authorisation by granting authorisations directly to keys (bypassing identities).

Other works (e.g., [GPSS05]) have also investigated solutions for providing authentication and access control based on biometry [GLM<sup>+</sup>04]. In this context, Cimato et al. [CGP<sup>+</sup>08] propose a privacy-aware biometric authentication technique that uses multiple biometric traits.

### ***16.4.2 Data Handling Policy Languages***

Some works have also focused on the definition of privacy policy languages [ACDS08, AHKS02, eXt05, W3C06a, Web06] that support preliminary solutions to the privacy protection issue, as for instance, by providing functionalities for controlling secondary use of data (i.e., how personal information could be managed once collected). The Platform for Privacy Preferences (P3P) [Cra02, W3C06a] is a World Wide Web Consortium (W3C) project that addresses the need of a client to assess whether the privacy practices adopted by a server comply with her privacy



preferences before the release of personal information (see also Section 19.3.2). To this aim, P3P provides an XML-based language and a mechanism for ensuring that clients can be informed about the privacy policies of the server. The corresponding language that would allow clients to specify their preferences as a set of rules is called *A P3P Preference Exchange Language* (APPEL) [W3C02]. Privacy preferences are specified and evaluated by the clients before data releases. SecPAL for Privacy (S4P) [BMB09] is a logic-based language to specify privacy policies and preferences. S4P specifies preferences as *may* assertions (i.e., authorisations) and *will* queries (i.e., obligation requests). S4P specifies policies as *will* assertions (i.e., commitments on obligations) and *may* queries (i.e., authorisation requests). The work in [ACDS08] presents a solution for secondary use management that integrates a credential-based access control system with data handling policy definition, evaluation, and enforcement. A data handling policy language provides the users with the possibility to specify data recipients, usage purposes, and obligations, thus regulating how their personal information can be subsequently used by external parties receiving it.

Data handling is sometimes also referred to as *usage control* [PS04, HBP05]. The Obligation Specification Language (OSL) [HPB<sup>+</sup>07] supports a wide range of usage control requirements related to time, cardinality, purpose, and events. OSL is logic-based, so that a sequence of events can automatically be checked for compliance with the specified policy. The OSL extensions for policy evolution [PSSW08] target a use case similar to the PrimeLife use case of downstream usage. However, in OSL it is the data provider who unilaterally creates the policy to be adhered to, whereas PrimeLife aimed to develop a language in which such a policy is the result of matching a consumer's policy against the provider's preferences.

Even if scenarios and trust models are different, there are clear similarities between digital rights management (DRM), enterprise rights management (ERM), and data handling policies. Indeed, in each case, a data provider attaches constraints, in the form of a license or a sticky policy, to data sent to a data consumer. The domain-specific vocabularies for privacy policies and rights expression languages (RELs) may be rather different, but the same overall language structure can be used for both. The state of the art ERM and DRM languages are MPEG-21 REL [Wan04], XrML [Con02], and ODRL [ODR02].

### 16.4.3 *Anonymous Credential Systems and Private Information Management*

Some effort has also been done in the context of anonymous credential systems [IDE].

An anonymous credential [CL01, Cha85] can be seen as a signed list of attribute-value pairs issued by a trusted issuer. They have the advantage that the owner can reveal only a subset of the attributes, or even merely prove that they satisfy some conditions, without revealing any more information about the other attributes. Also, they provide additional privacy guarantees like unlinkability, meaning that even

with the help of the issuer a server cannot link multiple visits by the same user to each other, or link a visit to the issuing of a credential. There are two main anonymous credential systems in use today, namely Identity Mixer [CL01, CV02] and U-Prove [U-P07]. Both are privacy-enhancing public-key infrastructures allowing users to selectively disclose attributes from their credentials and prove conditions over their attributes without revealing their full values. The U-Prove system provides one-time credentials, whereas Identity Mixer credentials can be used multiple times. Identity Mixer also has a number of interesting associated cryptographic tools, such as verifiable encryption [CD00] that permits proving properties about encrypted values, and limited spending [BCC05] that allows for restrictions to be placed on how often the same credential can be used to access a service, without compromising anonymity.

Existing languages are not targeted to anonymous transactions and thus lack the ability for expressing semantics for obtaining accountability in anonymous and unlinkable transactions, which can be achieved through the capability of disclosure to third parties. The latter is a crucial requirement when considering a practical language for data minimisation scenarios, such as anonymous transactions, particularly when considering the current legislation trend. The first paper towards third-party disclosure is by Backes et al. [BCS05]. In [GD06], P3P is extended such that it allows for describing credentials and their properties, which are necessary for gaining service access. The language is XML-based, and credential descriptions also allow for verifiable encryptions as a special case of third party attribute disclosure. A language featuring a credential typing mechanism and advanced features such as spending restrictions and signing requirements was recently proposed by Ardagna et al. [ACK<sup>+</sup>10].

Recent research on credential-based access control (e.g., [BS02, IY05, LWBW08, RZN<sup>+</sup>05, YWS03]) has focused on client side issues and proposed solutions for regulating the release of users' private information (also in the form of anonymous credentials) and possibly managing negotiation with the server. Chen et al. [CCKT05] propose a solution that associates costs with credentials and policies to minimise the cost of a credential release within a trust-negotiation protocol. Kärger et al. [KOB08] describe a logic-based language for the specification of privacy preferences dictating a partial order among the client properties. Both solutions provide a treatment of preferences or scores associated with either credentials or properties. Yao et al. [YFAT08] propose a point-based trust management model, where the client labels each credential in its portfolio with a quantitative privacy score, while the server defines a credit for each credential released by the client and a minimum threshold of credits to access a resource. The proposed solution finds an optimal set of client credentials, such that the total privacy score of disclosed credentials is minimal and the server access threshold is satisfied. Finally, Ardagna et al. [ADF<sup>+</sup>10a, ADF<sup>+</sup>10b] define solutions that permit the client to define its privacy preferences in terms of sensitivity labels of portfolio components and to minimise the disclosure of sensitive information, independently from the server preferences. These proposals provide a complete modeling of the client portfolio, consider emerging technologies such as anonymous credentials, and capture sensitive associations and disclosure

constraints of client properties and credentials. The research community has also focused on protecting the privacy of users once their data are released to and stored by servers. Several approaches have been presented including techniques based on  $k$ -anonymity and  $k$ -anonymous data mining. A summary of existing data protection techniques is presented in [CDFS07a, CDFS07b, CDFS08].

To conclude, existing solutions usually provide access control languages and solutions that are logic-based, powerful, highly expressive, and permit the specification of complex conditions involving credentials and relations between parties in a simple yet effective way. However, in real world scenarios, such as the one considered in PrimeLife, fundamental requirements for access control solutions are simplicity and ease of use, rather than the presence of a complete and highly expressive access control language. Also, although the benefits of all these works (e.g., credential integration), few of them provide functionalities for protecting the privacy of users and regulate the use of their personal information in secondary applications.

## Chapter 17

# Matching Privacy Policies and Preferences: Access Control, Obligations, Authorisations, and Downstream Usage

Laurent Bussard, Gregory Neven, and Franz-Stefan Preiss

**Abstract** This chapter describes how users' privacy preferences and services' privacy policies are matched in order to decide whether personal data can be shared with services. Matching has to take into account data handling, i.e. does services handle collected data in a suitable way according to user expectations, and access control, i.e. do the service that will be granted access to the data comply with user expectations. Whereas access control describes the conditions that have to be fulfilled before data is released, data handling describes how the data has to be treated after it is released.

Data handling is specified as obligations that must be fulfilled by the service and authorisations that may be used by the service. An important aspect of authorisation, especially in light of the current trend towards composed web services (so-called mash-ups), is downstream usage, i.e., with whom and under which data handling restrictions data can be shared.

## 17.1 Privacy Specifications: Preferences, Policies, and Sticky Policies

The scenario we consider is one where two parties, typically a user and a server, engage in an interaction where one of the parties, typically the server, requests some personal data, e.g., personally identifiable information (PII), from the other party (See [Figure 17.1](#)). We will from now on call the party that provides the data the *data subject* and the party that requests the data the *data controller*. Moreover, we consider a scenario where, at a later point in time, the data controller may want to forward personal data to a third party, called the *downstream data controller*.

Both the data subject and the data controller have their own policies expressing the required and proposed treatment of personal data, respectively. These policies contain access control and data handling requirements. Personal data are only sent to a data controller after (1) the access control requirements have been met, and (2)

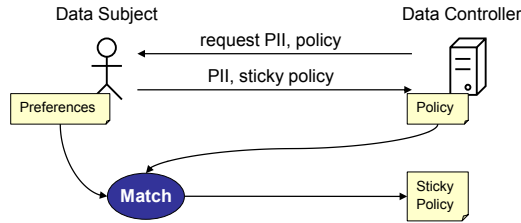


Fig. 17.1: Matching data subject's privacy preferences and data controller's privacy policy.

a suitable data handling policy has been agreed upon. We distinguish three kinds of policies:

**Preferences:** In his *preferences*, the data subject describes, for specific pieces of personal data, which access control requirements a data controller has to satisfy in order to obtain this personal data, as well as the data handling requirements according to which personal data has to be treated after transmission. These requirements may include *downstream usage requirements*, meaning the requirements that a downstream data controller has to fulfill in order to obtain personal data from the (primary) data controller.

**Policy:** The *policy* is the data controller's counterpart of the data subject's preferences. In a policy the data controller defines, for specific pieces of personal data to be obtained, his certified properties (roles, certificates, etc.) that can be used to fulfill access control requirements, and a data handling policy describing how he intends to use personal data.

**Sticky policy:** The *sticky policy* describes the mutual agreement concerning the usage of a transmitted piece of personal data. This agreement is the result of a matching process between a data subject's preferences and a data controller's policy. Technically a sticky policy is quite similar to preferences as described above, but it describes a mutual agreement between the data subject and the data controller that cannot be changed. After receiving personal data, the data controller is responsible for storing and enforcing the sticky policy.

Figure 17.2 provides the overall structure of the language. A similar structure is used to specify policies, preferences, and sticky policies.

*Applicability* specifies which personal data are targeted by the policy. *Applicability* in preferences specifies the type of data (or a specific data) targeted by the preference. *Applicability* in policies defines which parameter (type of data collected through a given interface) is targeted by the policy. *Applicability* is not part of sticky policies.

*ACUC* groups access control and data handling. *AccessControl* defines claims required to gain access to the personal data. *AccessControl* in preferences defines properties of services that can gain access under this preference. *AccessControl* in

```

<Policies>
  <Policy>
    <Applicability> ... </Applicability>
    <ACUC>
      <AccessControl> ... </AccessControl>
      <UsageControl>
        <Rights> ... </Rights>
        <Obligations> ... </Obligations>
      </UsageControl>
    </ACUC>
  </Policy>
  ...
</Policies>

```

Fig. 17.2: Structure of policies.

policies defines properties of the service exposing the policy. *AccessControl* is not defined in sticky policies.

*UsageControl*<sup>1</sup> specifies how data is handled. It defines *Rights*, i.e., what the service is authorised to do with the data, and *Obligations*, i.e., obligations that must be fulfilled by the service. Preference-side *Rights* are the rights the user is willing to grant in a specific context. Policy-side *Rights* are the rights required by the service for a specific collected data. Preference-side *Obligations* are the obligations required by the user in a specific context. Policy-side *Obligations* are the obligations the service is willing to fulfill for a specific collected data.

## 17.2 Matching Data Handling

Given a data subject's preferences and a controller's policies, matching aims at automating the process of deciding whether the data subject can safely transmit a piece of personal data. We introduce a '*more or equally permissive than*' operator ( $\triangleright$ ) to match preferences with policies. Intuitively, more permissive means more rights and/or less obligations. We say there is a *match* when the preferences are more or equally permissive than the policy.

### 17.2.1 Boolean Match

Matching preferences and policies boils down to matching individual rights and obligations. Data controllers generally expose an interface (e.g. HTML Forms or WSDL) specifying the type of requested personal data. *Applicability* is used to as-

<sup>1</sup> In this chapter we consider *usage control* (UC) and *data handling* (DH) as synonyms.

sociate each parameter  $p$  of this interface with one privacy policy  $Pol_p$ . On the data subject side, for each possible personal data  $pii$ , *applicability* determines the set of relevant privacy preferences  $Prefs_{pii}$ . When  $Prefs_{pii} \supseteq Pol_p$ , the assignment  $p \leftarrow pii$  does match. More precisely, a set of preferences is matched with a policy as follows:

$$Prefs \supseteq Pol \Leftrightarrow \exists Pref \in Prefs \cdot (Pref.ACUC \supseteq Pol.ACUC) \quad (17.1)$$

When multiple assignments are possible, e.g. data controller requires an e-mail address  $p$  and data subject has a corporate address  $pii_0$  and a private address  $pii_1$ , multiple matches  $Prefs_{pii_0} \supseteq Pol_p$  and  $Prefs_{pii_1} \supseteq Pol_p$  are simultaneously evaluated before picking an assignment, i.e. during identity/data selection.

In the following, we use the notations  $*_{Pref}$  and  $*_{Pol}$  to denote elements within preferences and policies respectively. Pairs of access control and data handling policies are matched as follows:

$$ACUC_{Pref} \supseteq ACUC_{Pol} \Leftrightarrow (ACUC_{Pref}.AC \supseteq ACUC_{Pol}.AC) \wedge (ACUC_{Pref}.UC \supseteq ACUC_{Pol}.UC) \quad (17.2)$$

Note that (17.2) is evaluated multiple times during the evaluation of (17.1). For example,  $ACUC_{Pref}$  is instantiated subsequently with  $Pref_i.ACUC$  for all  $Pref_i$  in  $Prefs$ . Matching Access Control policies is out of the scope of this chapter. When the data subject specifies rules (e.g. using XACML) and the data controller has attributes (e.g. X.509 or SAML), matching is implemented as an access control decision. Data handling requirements are matched as follows:

$$\begin{aligned} UC_{Pref} \supseteq UC_{Pol} \Leftrightarrow & \\ & (\forall R \in UC_{Pol}.Rights \cdot \exists R' \in UC_{Pref}.Rights \cdot R' \supseteq R) \wedge \\ & (\forall O \in UC_{Pref}.O \cdot \exists O' \in UC_{Pol}.O \cdot O \supseteq O') \end{aligned} \quad (17.3)$$

Matching individual obligations ( $O \supseteq O'$ ) is specified in Section 17.3. Matching individual authorisations ( $R' \supseteq R$ ) is defined in Sections 17.4 and 17.5.

## 17.2.2 Going Further than Boolean Match

Formulas presented in this chapter give an idea of the logic used to compare policies. However, the implementation is more complex because a Boolean result is generally not sufficient.

In case of a match, a sticky policy expressing the agreement between the data subject and the data controller has to be issued. When personal data  $pii$  is assigned

to parameter  $p$ , the resulting sticky policy  $SP_{p \leftarrow pii}$  must fulfill the following conditions:  $Prefs_{pii} \supseteq SP_{p \leftarrow pii} \supseteq Pol_p$ .

In case of a mismatch, more details are also required. First, the cause of the mismatch has to be identified. Second, remediation can be proposed in order to modify the preferences and get a match. In order to identify the source of the mismatch, and to propose a remediation, we use a mechanism to measure the similarity of pieces of policy. Similarity makes it possible for the user to choose between canceling the transaction and changing her preferences in a privacy-friendly way. When assigning personal data  $pii$  to parameter  $p$  leads to a mismatch ( $Prefs_{pii} \not\supseteq Pol_p$ ), new preferences  $Prefs'_{pii}$  may be proposed in order to have  $Prefs'_{pii} \supseteq Pol_p$  while minimising the differences between  $Prefs'_{pii}$  and  $Prefs_{pii}$ .

### 17.3 Obligations

It is not possible to develop an exhaustive list of obligations because too many domain-specific obligations can be envisioned, e.g. the obligation to notify the data subject's doctor in a health scenario. For this reason, we developed a set of usual obligations (e.g. data retention) and extension points for specifying upcoming or specific obligations. We define an obligation with one action and  $n$  triggers as:

Do **action** when { **trigger**<sub>1</sub>  $\vee$  **trigger**<sub>2</sub>  $\vee$  ...  $\vee$  **trigger** <sub>$n$</sub>  }

where *action* defines the action to execute to fulfill the obligation and *trigger* specifies the event and conditions requiring the execution of this action. When an obligation specifies multiple triggers, each event corresponding to one or more triggers will result in the execution of the action. Obligations cannot have multiple actions because rollback would be too complex. For instance, six months data retention is expressed as the obligation to “delete data within 6 months”:

Do **DeletePersonalData()** when { **AtTime**( $t, maxDelay$ ) }

where  $t$  is the current date and  $maxDelay$  is six months. To compare obligations, we define:

$$\begin{aligned}
 O_{Pref} \supseteq O_{Pol} &\Leftrightarrow \\
 &(\forall T \in O_{Pref}.triggers \cdot \exists T' \in O_{Pol}.triggers \cdot T \supseteq T') \wedge \\
 &(O_{Pref}.action \supseteq O_{Pol}.action)
 \end{aligned} \tag{17.4}$$

Sections 17.3.1 and 17.3.2 give more details on triggers and actions respectively.



### 17.3.1 Triggers

We defined seven triggers, namely *At Time*, *Periodic*, *Personal Data Accessed For Purpose*, *Personal Data Deleted*, *Personal Data Sent*, *Data Lost*, and *On Violation*.

Trigger “*TriggerAtTime*” is defined in Figure 17.3. This trigger has two parameters: “*Start*,” i.e. when the trigger may be started, and “*MaxDelay*,” i.e. the response time. In other words, this trigger is correctly enforced by triggering the associated action once between “*Start*” and “*Start + MaxDelay*”. For instance, “*delete within one year*” and “*delete next month*” are translated into *TriggerAtTime(now, 1year)* and *TriggerAtTime(x, y - x)* respectively where *x* is the first day of next month and *y* is the last day of next month.

```
<xs:complexType name="TriggerAtTime">
  <xs:complexContent>
    <xs:extension base="ob:Trigger">
      <xs:sequence>
        <xs:element name="Start" type="ob:DateAndTime" ... />
        <xs:element name="MaxDelay" type="ob:Duration" ... />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Fig. 17.3: Example of XML schema for trigger “*TriggerAtTime*.”

When evaluating triggers as specified in Formula 17.3, triggers of type “*TriggerAtTime*” are matched as follows.

$$\begin{aligned}
 T_{Pref} \supseteq T_{Pol} \Leftrightarrow & (Type(T_{Pref}) = Type(T_{Pol}) = TriggerAtTime) \wedge \\
 & (T_{Pref}.Start \leq T_{Pol}.Start) \wedge \\
 & ((T_{Pref}.Start + T_{Pref}.MaxDelay) \geq (T_{Pol}.Start + T_{Pol}.MaxDelay))
 \end{aligned}
 \tag{17.5}$$

In other words, one trigger is more permissive than another when it specifies less constraints. In case of “*TriggerAtTime*,”  $T_{Pref}$  is more permissive than  $T_{Pol}$  when it starts earlier and/or ends later. The obligation of deleting within one year is thus satisfied by the obligation of deleting within six months.

Trigger “*TriggerAccessedForPurpose*” leads to the execution of the associated action (within a *maximum delay*) after each use of personal data for one of the specified *purposes*. Such triggers are matched as follows.

$$\begin{aligned}
T_{Pref} \supseteq T_{Pol} \Leftrightarrow & (Type(T_{Pref}) = Type(T_{Pol}) = TriggerAccessedForPurpose) \wedge \\
& (T_{Pref}.Purposes \subseteq T_{Pol}.Purposes) \wedge \\
& (T_{Pref}.MaxDelay \geq T_{Pol}.MaxDelay)
\end{aligned} \tag{17.6}$$

In other words, one trigger “*TriggerAccessedForPurpose*” is more permissive than another when it reacts to fewer events (i.e. a subset of purposes) or slower (i.e. longer response time). The obligation of notifying each use for purpose “treatment” within one day is more permissive than the obligation of notifying any use within one hour.

Defining the syntax and semantics of all triggers is out of the scope of this chapter. Look at [Pri09b] for more details. Here is a short description of predefined triggers.

- *TriggerAtTime(start, maxDelay)*: executes the associated action once at some time between start and start + maxDelay.
- *TriggerPeriodic(start, end, maxDelay, period)*: executes the associated action once per period.
- *TriggerPersonalDataAccessedForPurpose(purpose, maxDelay)*: executes the associated action each time the personal data is used for specified purposes.
- *TriggerPersonalDataDeleted(maxDelay)*: executes the associated action when the personal data is deleted.
- *TriggerPersonalDataSent(thirdParty, maxDelay)*: executes the associated action when the personal data is shared with a third party.
- *TriggerDataLost(maxDelay)*: executes the associated action in case of a major issue leading to data theft.
- *TriggerOnViolation(obligation, maxDelay)*: executes the associated action in case of a violation of the referenced obligation.

### 17.3.2 Actions

We defined four actions, namely *Secure Log*, *Delete Personal Data*, *Anonymise Personal Data*, and *Notify Data Subject*.

Action “*ActionSecureLog*” is defined in Figure 17.4. This action has five parameters: “*Integrity Level*,” i.e. protection against modification of logs, “*Confidentiality Level*,” i.e. protection against unauthorised accesses of logs, “*Non-Repudiation Level*,” i.e. protection against repudiation of logs, “*Time-Stamping Level*,” i.e. guarantees on when a log was added, and “*Availability Level*,” i.e. protection against lost of logs.

When evaluating actions as specified in Formula 17.3, actions of type “*ActionSecureLog*” are matched as follows:

```

<xs:complexType name="ActionSecureLog">
  <xs:complexContent>
    <xs:extension base="ob:Action">
      <xs:sequence>
        <xs:element name="IntegrLevel" type="xs:decimal" ... />
        <xs:element name="ConfidLevel" type="xs:decimal" ... />
        <xs:element name="NonRepLevel" type="xs:decimal" ... />
        <xs:element name="TimeStampLevel" type="xs:decimal" ... />
        <xs:element name="AvailabLevel" type="xs:decimal" ... />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Fig. 17.4: Example of XML schema for action “ActionSecureLog.”

$$\begin{aligned}
A_{Pref} \supseteq A_{Pol} \Leftrightarrow & (Type(A_{Pref}) = Type(A_{Pol}) = ActionSecureLog) \wedge \\
& (A_{Pref}.IntegrLevel \leq A_{Pol}.IntegrLevel) \wedge \\
& (A_{Pref}.ConfLevel \leq A_{Pol}.ConfLevel) \wedge \\
& (A_{Pref}.NonRepLevel \leq A_{Pol}.NonRepLevel) \wedge \\
& (A_{Pref}.TimeStampLevel \leq A_{Pol}.TimeStampLevel) \wedge \\
& (A_{Pref}.AvailabLevel \leq A_{Pol}.AvailabLevel) \quad (17.7)
\end{aligned}$$

In other words, one action is more permissive than another when it specifies fewer constraints. In case of “ActionSecureLog,”  $A_{Pref}$  is more permissive than  $A_{Pol}$  when it requires less security properties in terms of integrity, confidentiality, non-repudiation of origin, time-stamping, and/or availability.

Defining the syntax and semantics of all actions is out of the scope of this chapter. Look at [Pri09b] for more details. Here is a short description of predefined actions.

- ActionSecureLog(integrityLevel, confidentialityLevel, nonRepudiationLevel, timeStampingLevel, availabilityLevel): the action of logging specific events related to personal data.
- ActionDeletePersonalData: the action of deleting personal data.
- ActionAnonymizePersonalData: the action of removing identifiers from personal data
- ActionNotifyDataSubject(media, address): the action of notifying data subject about specific events related to personal data.

### 17.3.3 Enforcement

The “Obligation Enforcement Engine” is in charge of enforcing obligations. When personal data are collected, related triggers are analysed in order to register for rel-

event events (e.g. access to personal data stored in a legacy database) and schedule future actions (e.g. schedule a deletion in eleven months to enforce “delete within one year”). The enforcement engine reacts to relevant events and executes associated actions (e.g. log, delete).

## 17.4 Authorisations

When occurring in a Data Controller’s policy, authorisations express the minimal (i.e., the least permissive set of) rights that a Data Controller wants to obtain on requested personal data. When occurring in a Data Subject’s preferences, they express the maximal (i.e., the most permissive set of) rights that she is willing to grant to a Data Controller with respect to her personal data. When occurring in a sticky policy, authorisations express the rights that the Data Subject has explicitly agreed to grant to the Data Controller.

The main difference with obligations is that *not* performing an authorised action is not a violation of the policy. Performing an action that is not explicitly authorised, on the other hand, is a violation.

We model two types of authorisations, namely the authorisation to use personal data for a specified purpose, and the authorisation to forward personal data to third parties (i.e., downstream usage). We focus on the former type here, and discuss the latter type in more detail in Section 17.5.

Authorisation *UseForPurpose* takes a single parameter  $p$ , which is a string indicating the purpose for which personal data is to be used. Two *UseForPurpose* authorisations match whenever their purposes are equal:

$$UseForPurpose(p) \supseteq UseForPurpose(p') \Leftrightarrow p = p'.$$

As an extension, one could see usage purposes as occurring in a hierarchy, rather than as a flat list, so that for example “telemarketing” can be a subpurpose of “marketing.” The matching definition then needs to be adapted so that a match occurs whenever  $p$  is an ancestor of or equal to  $p'$ .

The authorisation to use personal data only for a specified set of purposes can be enforced in the Data Controller’s infrastructure by annotating each access request to personal data with the intended usage purpose, and by setting the access control policy of personal data so that only requests for purposes included in the sticky policy will be permitted.

## 17.5 Downstream Data Handling

A second authorisation that we model in our vocabulary is the authorisation to forward personal data to other Data Controllers, or as we call it, *downstream usage* of

personal data. In her preferences, the Data Subject specifies to which downstream Data Controllers her personal data can be forwarded, and which data handling policy these downstream controllers have to adhere to. For the Data Controller, we specify two different mechanisms to express his intentions of forwarding personal data downstream. Which mechanism is chosen also affects the matching algorithm, as we will see below.

Our language supports nested downstream usage policies, allowing Data Subjects, as well as Data Controllers, to specify in full detail the paths that personal data is allowed, or intended, to follow. It also allows recursion, enabling Data Subjects and Data Controllers to express restrictions under which personal data can be forwarded indefinitely. Optionally, a maximum forwarding depth can be specified.

### 17.5.1 Structure of Downstream Authorisations

The authorisation to forward the data is expressed by a *UseDownstream* element, which contains the ACUC restrictions under which it can be forwarded as a child *ACUC* element. This child *ACUC* element can in turn contain one or more *UseDownstream* elements, thereby enabling the specification of nested policies. If the *ACUC* element does not contain any nested downstream usage authorisations, then an optional parameter *maxDepth* can be used to indicate that personal data can be forwarded recursively under the restrictions of ACUC up to the indicated depth, which could be any integer or “unbounded”. The XML schema definition is given below:

```
<xs:element name="UseDownstream">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ACUC" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="maxDepth" type="int_or_unbounded"/>
  </xs:complexType>
</xs:element>
```

Let  $UseDownstream(ACUC)$  denote the authorisation to forward personal data under the restrictions of ACUC, and let  $UseDownstream(ACUC, depth)$  denote an authorisation to forward recursively up to a recursion depth *depth*. For a given graph of ACUCs, let  $|ACUC|$  be the “local” ACUC, meaning containing only those restrictions and obligations that do not affect downstream usage.

Using this notation, we can represent the structure of an ACUC policy with downstream usage as a directed graph where each node represents a hop in the downstream usage. Each node is labeled with the local ACUC policy describing how the data are to be treated locally. Each edge represents the permission (in case of a provider’s preferences) or intention (in case of a consumer’s policy) to forward the data under the restrictions specified by the ACUC policy at the endpoint of the edge. For instance, the case where  $ACUC_A$  permits the right to share downstream under  $ACUC_B$ , but prohibits any further forwarding is depicted in [Figure 17.5\(a\)](#).

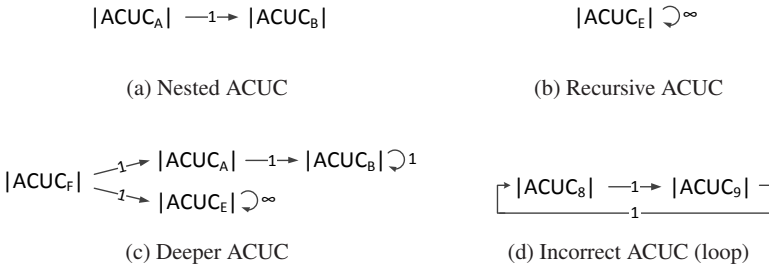


Fig. 17.5: Examples nested and recursive downstream usage.

By the restrictions that we imposed on recursion, the structure of the graph is similar to that of a tree where the leaf nodes can optionally have a loop, labeled with the maximal recursion depth. Figure 17.5(b), for example, represents a simple recursive ACUC. Figure 17.5(c) shows a more complicated nested structure, where  $ACUC_F$  specifies that the data can either be forwarded indefinitely  $ACUC_E$ , or once under  $ACUC_A$  and twice under  $ACUC_B$ . Figure 17.5(d) is not valid, however, because it contains a cycle. For the sake of readability and to avoid complicating the matching procedure, we explicitly forbid cycles other than simple recursions.

### 17.5.2 Proactive Matching of Downstream Data Handling

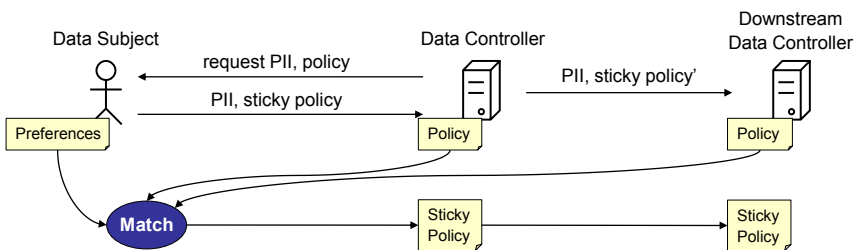


Fig. 17.6: Proactive matching of data subject's privacy preferences and privacy policies of data controllers.

For the first matching mechanism, which we call *proactive matching*, the Data Controller's policy specifies in full detail to which downstream Data Controllers he intends to forward personal data, and how they will treat it. Optionally, the Data

Controller's policy can point to remote downstream policies hosted by the downstream Data Controllers themselves. Either way, proactive matching requires that the full chain of downstream Data Controllers and their policies are known at the moment that the Data Subject releases her personal data to the first Data Controller, so that the entire policy chain can be taken into account by the matching algorithm to reach a decision, as depicted in [Figure 17.6](#).

The advantage of proactive matching is that the policies of all downstream Data Controllers are known to the Data Subject at the moment she releases her personal data to the first Data Controller. This allows her to make a better informed decision on whether or not to reveal her personal data, and, in case of a mismatch between her preferences and a downstream controller's policy, gives her the option to consciously overrule her own preferences. A possible disadvantage is that the Data Controller's full workflow has to be known and fixed at the moment that personal data is released. Not only could the workflow leak sensitive information about the Data Controller's business processes, but it is also not clear what happens when a downstream controller's policy changes between the moment that personal data is first revealed and the moment that it is forwarded. Such scenarios require another matching algorithm: "lazy matching" (See Section 17.5.3).

Matching non-recursive (but possibly nested) downstream usage authorisations is done by checking whether the specified ACUC restrictions match:

$$UseDownstream(ACUC) \supseteq UseDownstream(ACUC') \Leftrightarrow ACUC \supseteq ACUC' .$$

Two recursive downstream usage authorisations are matched by additionally checking the recursion depths:

$$\begin{aligned} UseDownstream(ACUC, depth) \supseteq UseDownstream(ACUC', depth') \\ \Leftrightarrow ACUC \supseteq ACUC' \wedge depth \geq depth' . \end{aligned}$$

Recursive and non-recursive downstream authorisations are essentially matched by "folding out" both recursion trees and simultaneously iterating over the nodes in the two tree representations to verify that it is possible to cover each branch of the policy-side tree with a more or equally permissive branch on the preference side. For instance, if in [Figures 17.5a](#) and [17.5b](#)  $|ACUC_E| \supseteq |ACUC_A|$  and  $|ACUC_E| \supseteq |ACUC_B|$ , then  $ACUC_E \supseteq ACUC_A$ . However, it is impossible that  $ACUC_A \supseteq ACUC_E$  because  $ACUC_E$  allows deeper recursive downstream usage than  $ACUC_A$ .

### 17.5.3 Lazy Matching of Downstream Data Handling

The proactive matching mechanism described above assumes that all policies of downstream Data Controllers are known at the time that the Data Subject releases her personal data to the first Data Controller. There may be situations, however, where it is not possible to collect all the necessary policies at this time, either be-

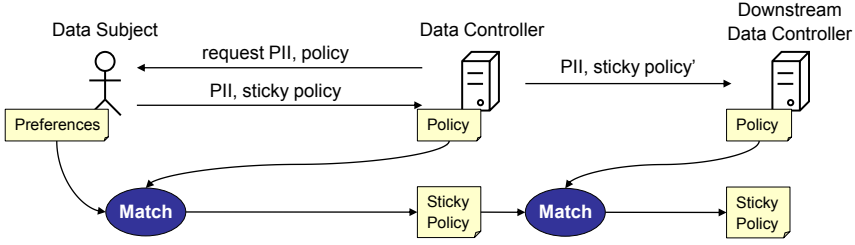


Fig. 17.7: Lazy matching of data subject's privacy preferences and privacy policies of data controllers.

cause the workflow reveals business secrets of the Data Controller, because the workflow is too complex to process efficiently, or because the downstream controllers are not known yet at the time of matching.

For this reason, we introduce a second mechanism called *lazy matching*, which only takes into account the properties and policies of the Data Controller, but not those of any downstream controllers. Here, the Data Controller's policy merely specifies whether he intends to forward personal data downstream. By declaring his intention to do so, he implicitly declares to be willing to impose any access and usage restrictions on the downstream Data Controllers that the Data Subject may specify. At the moment personal data is further forwarded, the downstream Data Controller's policy is matched against the sticky policy, which contains the Data Subject's preferences with regards to whom and under which conditions the data can be forwarded. This procedure is illustrated in Figure 17.7.

Both matching mechanisms ensure that eventually the Data Subject's preferences will be adhered to. Proactive matching is more privacy-friendly in the sense that it only gives away those authorisations that the Data Controllers explicitly applied for in their policies. Lazy matching has the advantage that if the downstream controllers' policies change between the moment of revealing and the moment of forwarding personal data, and the new policies are still within the Data Subject's preferences, the transaction can still go through, while it would have failed if proactive matching were used.

Since lazy matching gives the Data Subject slightly less control over her personal data, we introduce an additional boolean attribute *allowLazy* in an ACUC element by means of which the Data Subject can indicate in her preferences whether lazy matching is allowed for this ACUC policy. On the Data Controller's side, the attribute indicates whether he is willing to use lazy matching, and therefore to enforce any ACUC policy dictated by the Data Subject. Two such authorisations are matched according to the rule

$$\begin{aligned}
 UseDownstream(ACUC, lazy) &\supseteq UseDownstream(ACUC', lazy') \\
 &\Leftrightarrow (lazy \wedge lazy') \vee (ACUC \supseteq ACUC') .
 \end{aligned}$$



In the case that  $lazy \wedge lazy'$ , the resulting sticky policy will specify  $ACUC$  as the downstream usage policy, thereby ignoring  $ACUC'$  if it was present at all.

## 17.6 Conclusion

In this chapter, we presented a simple yet highly expressive language to specify privacy policies and preferences. It gives a clear view on the somewhat complex relation between access control and data handling policies, especially in the case where downstream usage is taken into consideration. We presented two strategies to match a Data Subject's preferences against Data Controllers' policies: proactive matching, which takes the full chain of downstream Controllers and their policies into account at the moment that personal data is revealed, and lazy matching, where the downstream policies are only matched when personal data is forwarded.

The policy engine implemented in PrimeLife (see Chapter 20) opted for the lazy matching algorithm. The main reason was that instead of the simple access control language used in this chapter, the PrimeLife Policy Language is embedded in the much more expressive industry standard XACML. In order to implement proactive matching, one would have to implement an algorithm that can test whether one XACML policy is more permissive than (i.e., is implied by) another XACML policy. Given the lack of formal foundations of XACML, this would be a considerable research effort in itself, which, because of the only marginal link to privacy, was deemed out of scope for PrimeLife.

## Chapter 18

# Advances in Access Control Policies

Claudio A. Ardagna, Sabrina De Capitani Di Vimercati, Gregory Neven, Stefano Paraboschi, Eros Pedrini, Franz-Stefan Preiss, Pierangela Samarati, and Mario Verdicchio

**Abstract** This chapter presents the results of the research on how the current standards for access control policies can be extended. In particular, Section 18.1 illustrates how privacy issues can be effectively tackled by means of a credential-based access control that includes anonymous credentials. Section 18.2 shows how the expressivity of policy languages can be exploited to introduce ontologies that model credential taxonomies and the relations among them, with a particular stress on the support for delegation mechanisms. Section 18.3 investigates the privacy issues that arise in those access control systems that are enriched with a dialog framework that enables servers to publish their policies. Finally, Section 18.4 maps these proposals onto a set of possible extensions of the architecture of the current de facto standard in access control policy languages: XACML.

### 18.1 Privacy-Preserving Access Control

Users commonly reveal much more personal data than necessary to be granted access to online resources, even though existing technologies offer functionalities that would allow for the authorisation to take place in a privacy-preserving way. The basic idea to achieve privacy-preserving access control is to utilise the cryptographic features of anonymous credential systems [CL01]. The concept of a credential as it will be used in the following is simply a *bundle* of attribute-value pairs that is signed by its issuer.

In our model of privacy-preserving access control systems [CMN<sup>+</sup>10], the decision as to whether access is granted to a requester is then dependent on the possession of, possibly multiple, credentials that fulfill certain requirements specified in a service provider's access control policy. For a user to obtain access to a protected resource, she produces a verifiable *claim* that contains cryptographic evidence that the policy is fulfilled and sends it to the service provider who verifies it with respect to his policy.

Basing the access control decision on the possession of credentials is not new. However, the use of anonymous credentials provides support to the following new features: (a) *predicate proofs* over attributes from – possibly *multiple* – credentials, (b) *selective* disclosure of *individual* attributes – possibly to third parties, and (c) the possibility to use predicate proofs and selective disclosure without interaction with the credential issuer. Only in access control systems that offer at least those features can the authorisation take place in a truly privacy-preserving way, since a user can thus control which data and predicates are released and which are not.

We believe that there are various reasons for not adopting anonymous credentials for authorisation systems. One reason is the fact that industry is currently more interested in the possibility of profiling their users than in protecting their users' privacy. However, taking into account that privacy violations are reported by the media on an almost daily basis, it seems that public awareness on that topic is rising slowly but steadily and that the industry will need to adapt to these changes in the near future. Another reason for this lack of technology adoption is the absence of a suitable authorisation language offering adequate expressiveness to address the privacy-friendly functionalities of anonymous credentials.

To overcome the latter problem, we have developed an authorisation language that allows for expressing access control requirements in a privacy-preserving way. Although our language is targeted towards anonymous credentials, it allows for the specification of authorisation requirements regardless of the underlying technology and its implementation details, and it is also applicable for credential technologies designed without privacy considerations.

Before we describe our language and its features, we give a more detailed overview of credentials and their functionalities. Afterwards, we describe by means of a comprehensive example policy, how the credential functionalities are mapped into our language.

### ***18.1.1 Credentials Enabling Privacy-Preservation***

A credential is a bundle of attribute-value pairs that is provided by an issuer to an individual that becomes the credential's owner. Credentials are always of a certain *type* that specifies the list of attributes that a credential contains. For example, a national ID card (issued by a government) may contain the first name, last name and date of birth of the owner, while a movie ticket (issued by a movie theater) contains the time and date of the showing and a seat number. The issuer vouches for the correctness of the information on the credential with respect to the intended owner. The issuing process may be carried out on-line, e.g., by visiting the issuer's website, as well as off-line, e.g., at the local town hall. Credentials are issued by means of a certain credential technology. Technologies that support our credential model are, e.g., anonymous credentials [CL01], X.509 certificate [CSF<sup>+</sup>08], OpenID [Ope07] or SAML [OAS05a].

To gain access to a resource protected by a policy, the server has to be convinced that the policy is fulfilled. To do so in our model, the credential owner makes a *claim* about the credentials she owns and about the attributes they contain. Although claims are made independently from any concrete technology, the accompanying claim *evidence* that authenticates them (typically generated by means of cryptographic mechanisms) is specific to the technology underlying the credentials.

In the following, we list a number of features that existing credential technologies have:

- **Proof of ownership.** To bind a credential to its legitimate owner, a credential may contain information that is used to authenticate the owner. This could be a picture of the user, a PIN code, a password, or a signing key. Proving credential ownership means that the owner authentication is successfully performed with whatever mechanism is in place.
- **Selective attribute disclosure.** Some technologies allow attributes within a credential to be revealed selectively, i.e., the service provider only learns the value of a subset of the attributes contained in the credentials.
- **Proving conditions on attributes.** Anonymous credentials enable proving conditions over attributes *without* revealing their actual values. For all other technologies, the only way to prove that an attribute satisfies a condition is by revealing its value.
- **Attribute disclosure to third parties.** Attributes are usually revealed to the relying party enforcing the policy, but the policy could also require certain attributes to be revealed to an external third party. For example, the server may require that the user reveals her full name to a trusted escrow agent, so that she can be de-anonymised in case of fraud, thereby adding accountability to otherwise anonymous transactions.
- **Signing of statements.** Certain technologies allow for the signing of a given statement to explicitly consent to it. The signature acts as evidence that this statement was agreed to by a user fulfilling the policy in question.

We regard credential technologies that support (at least) the first three of the above mentioned features as *privacy-preserving* credential technologies. The utilisation of such technologies enables us to perform access control in a privacy-preserving way, which means, in the ideal case, that no more information than strictly required is revealed to fulfill a policy. As mentioned earlier, in the context of privacy-preserving technology, our focus is on anonymous credentials.

### 18.1.2 A Policy Language for Privacy-Preserving Access Control

The credential-based access control requirements language (CARL) that we have developed allows service providers to express the requirements that a user's credentials have to satisfy in order to gain access to a resource. This expressivity is achieved with the following features:

- **Privacy preservation.** Our language is privacy-preserving in the sense that it supports the principle of minimal information disclosure, i.e., a policy expresses the minimal claim that a user has to present. It does so in terms of which credentials have to be involved in the claim, which attributes of those credentials have to be revealed, and which conditions have to hold over the attributes (no matter whether these are revealed or not). Rather than assuming that all attributes in a credential are revealed by default, it clearly distinguishes between the requirement to reveal the value of an attribute (e.g., the date of birth) and the requirement that an attribute has to satisfy a certain condition (e.g., age greater than 18). This approach allows the user to minimise the amount of data that she reveals to the server, which is important as credentials often contain sensitive personal information. Additionally, our language also supports *accountability*, so that the user's anonymity can be revoked by a third party in case of abuse.
- **Technology independence.** Our language is independent of the technology underlying the credentials, so that different technologies or even a mix of technologies can be used without modifying the policy specifications. Thus, service providers can specify policies without having to worry about the specifics of the underlying credential technology.
- **Multi-credential claims.** Our policy language can express requirements involving multiple credentials at the same time and has a way to refer to individual credentials and the attributes they contain. It can thereby impose "cross-credential" conditions, i.e., conditions involving attributes from different credentials. The possibility to reference individual credentials is also important when a user has multiple credentials of the same type. For example, when a user has two credit cards, the policy should be unambiguous about whether it wants to see the credit card number and security code of the *same* card or of *different* cards.

Here follows a comprehensive example of our policy language that captures all of its aspects and main features about a policy that is used by a car rental company to determine eligibility for a *discounted* rental car:

- 01: own *mc::MembershipCard* issued-by CARRENTALCOMPNAV
- 02: own *cc::CreditCard* issued-by AMEX, VISA
- 03: own *dl::DriversLicense* issued-by DEPTOFMOTORVEHICLES
- 04: own *is::LiabilityInsuranceStmt* issued-by INSURANCECOMPANY
- 05: reveal *cc.number* under 'purpose=payment'
- 06: reveal *is.policyNo* to ESCROWAGENT under 'in case of damage'
- 07: sign 'I agree with the general terms and conditions.'
- 08: where *dl.vehicleCategory* = 'M1'  $\wedge$  *is.guaranteedEURAmount*  $\geq$  '30.000'  $\wedge$
- 09:       (*mc.status* = 'gold'  $\vee$  *mc.status* = 'silver')  $\wedge$  *cc.expDate*  $>$  *today()*  $\wedge$
- 10:       *mc.name* = *dl.name*  $\wedge$  *dl.name* = *is.name*

The policy states that users are eligible who (1) have a membership status of *gold* or *silver* with the company, (2) reveal the number of a *valid* American Express or Visa credit card for payment purposes, (3) are entitled by the Departement of Motor Vehicles to drive passenger vehicles, (4) reveal the insurance policy number of a liability insurance with coverage of at least thirty thousand Euros to a trusted escrow

agent who may disclose this policy number only in case of damage, (5) consent to the general terms and conditions, and (6) have the membership card, the drivers license and the insurance statement issued on the same name (to ensure that the driving person is a member and insured). Through the use of credential identifiers, it is ensured that, e.g., the credit card number that is revealed must come from the same card on which the validity is tested.

An important aspect to note is that a user fulfilling this particular policy with privacy-preserving credentials does only reveal two pieces of information to the car rental company: the credit card number as well as the fact that she fulfills the policy. This comprises the assurance that the insurance policy number is revealed to the trusted escrow agent, which makes the user accountable in case of damage to the car. When revealing a credential's attribute, a data handling policy can also be attached to the revealed data: in the policy above, for instance, a purpose is attached to the request for the disclosure of the credit card number.

We have also defined the full grammar as well as the formal semantics for our policy language [CMN<sup>+</sup>10]. The semantics abstractly defines the intended behaviour of an access control system for a given policy and thereby defines the obligations that an actual implementation must meet.

## 18.2 Credential Ontologies: Concepts and Relations

The formal language in logic-based proposals, including ours, can be extended to perform ontological inference and allow for the derivation of new concepts (also known as abstractions, abbreviations, or macros) from an initial set of basic concepts.

Abstractions represent a shorthand for expressing, with a single concept, a composition (e.g. a set, a disjunction, a conjunction) of multiple concepts. Thus, the use of abstractions in the policy specification provides a compact and easy way to refer to complex concepts. For instance, “Id\_Document” can be defined as an abstraction for any element in a set of credentials like {Identity\_Card, Driver\_License, Passport}. An authorization specifying that the requester needs to provide an id-document to access a resource can then be satisfied with any of the four credentials above. A de facto standard like XACML does not provide explicit support for the creation of abstractions. Here follows a proposal of formal definitions as guidelines for such an extension.

### 18.2.1 Abstractions

Formally, we define an abstraction as follows.

**Definition 18.1 (Abstraction).** An abstraction is a rule of the form  $g \leftarrow d$ , where  $d$  is a complex credential condition, and  $g$  is a sequence of symbols that works as a meaningful shorthand.

For instance, abstractions:

- $c::Id\_Document \leftarrow c::Identity\_Card \vee c::Passport \vee c::Driver\_License$
- $e::E\_Money \leftarrow e::Credit\_Card \vee e::Debit\_Card \vee e::PayPal$

define *Id\_Document* and *E\_Money* as two abstract credential types corresponding to any element in the sets of credentials  $\{Identity\_Card, Passport, Driver\_License\}$  and  $\{Credit\_Card, Debit\_Card, PayPal\}$ , respectively. Hence, a request for an identifying document (credential of type *Id\_Document*) can be satisfied by providing either an identity card, a passport, or a driver license; a transaction calling for a payment with *E\_Money* can be carried out by means of a credit card, a debit card, or PayPal.

Abstractions can be exploited for defining and organizing concepts and taxonomies without the need for hierarchical data structures in traditional ontologies. Moreover, as shown in the following, abstractions can allow for the expression of policies based on chains of credentials, thus providing a support for the introduction of delegation mechanisms in credential-based systems.

## 18.2.2 Delegation by Recursion

One of the most interesting features offered by logic-based policy languages is that their expressiveness can be exploited to support recursive conditions [ADP<sup>+</sup>10]. Recursion plays a crucial role in the context of access control, as it allows us to express restrictions on how authorities and trusted parties can delegate the ability to issue credentials. The delegation consists of a certification of the ability of a third party to produce credentials on behalf of the delegator. Delegation increases the flexibility in complex distributed systems, and it allows for inexpensive creation of credentials, particularly in an open environment, where we often deal with application requirements calling for the specification of restrictions in delegation.

A policy language can support recursion by expressing conditions on data with a recursive structure. Let us illustrate this feature below.

Let  $U$  be the set of all users that can take part in an access control process. Let  $\rho \subseteq U \times U$  be a relation between elements in  $U$ . As an example to illustrate our ideas, let us focus on certification authorities, and let  $\rho$  be a relation that holds between two certification authorities  $u$  and  $v$  if and only if  $u$  has signed  $v$ 's public key on a certificate. With such signature,  $u$  delegates to  $v$  the authority to produce credentials, that is, any document certified by  $v$  is to be considered as certified by  $u$ . In turn,  $v$  has the possibility to delegate her power to another certification authority, so that a chain of delegation is created. The description of such organization must be maintained in a data structure accessible by all the users that rely on the relevant certification authorities.

Let  $\Theta_\rho$  be the information entity that exhaustively describes  $\rho$ , in the form of a sequence of entries modeled like credentials of type  $Rel_\rho$ , including two identifiers as attributes that refer to pairs of users that are in relation  $\rho$  in  $U$ :

$$\Theta_\rho = \{\theta :: Rel_\rho \text{ such that } (\theta.user_1, \theta.user_2) \in \rho\}.$$

When  $\rho$  holds between  $u$  and  $v$ , that is,  $(u, v) \in \rho$ , we assume that there exists a  $\theta \in \Theta_\rho$  such that  $\theta.user_1 = u$  and  $\theta.user_2 = v$ .

Conditions on data with a recursive structure like the one mentioned above can be requested in an access control policy. In our example, in which  $\rho$  is a relation of delegation between certification authorities, a requester trying to access a particular resource may be required by the server to show that the certification authority  $ca_r$  signing her credentials has been delegated by a particular authority  $ca_s$  preferred by the server.

The policy will then include the relevant condition on such a delegation:

- 01: own  $th :: Rel_\rho$  issued-by CA\_S
- 02: where  $th.user_1 = 'ca_s' \wedge th.user_2 = 'ca_r'$

Such condition can be easily rewritten according to the following abstraction:

$$th.users = \langle u, v \rangle \leftarrow th.user_1 = u \wedge th.user_2 = v$$

In this scenario, it may be convenient to introduce the transitive closure of the delegation chain. For instance, instead of setting conditions on the delegating authority that allowed for the requester's certificate to be certified by a third party, a server may be interested in ensuring that the root authority  $ca_{root}$  at the very beginning of the delegation chain is among her preferred ones.

The requester can prove that her  $ca_r$  is in the relation  $\rho^*$  (i.e., the transitive closure of  $\rho$ ) with  $ca_{root}$  either by showing that  $(ca_{root}, ca_c) \in \rho$ , or by providing a chain of context entries  $\theta_1, \dots, \theta_n \in \Theta_\rho$ , where  $ca_{root}$  is  $user_1$  in  $\theta_1$ ,  $ca_c$  is  $user_2$  in  $\theta_n$ , and for all  $1 \leq i < n$ ,  $\theta_i.user_2 = \theta_{i+1}.user_1$ , which can be abbreviated in what we define as a *recursive condition*:

$$th.users =^* \langle u, v \rangle \leftarrow th.users = \langle u, v \rangle \vee \\ th.users = \langle u, th'.user_1 \rangle \wedge th'.users =^* \langle th'.user_2, v \rangle.$$

### 18.3 Dialog Management

As mentioned before, the most widespread access control policy language to date, XACML, is affected by serious privacy issues.

In fact, it assumes that the engine enforcing access control has all the information needed to evaluate whether an authorization policy is satisfied, and no infrastructure to manage the dialog between the parties is called for. The evaluation of a policy can result in four possibilities: permit, deny, not applicable (if the policy does not apply to the request), or indeterminate (if the server does not have the information necessary to evaluate the policy). In an open world scenario, this would require the requester to reveal all the necessary credentials together with the service request.



The disclosure of the complete portfolio of credentials of a user is a very strong requirement: requesters reasonably want the ability to send to the counterpart only just what is needed to acquire access to the desired resources. Our proposal based on anonymous credentials enables access requests in accordance with the minimal disclosure principle. Such innovation, however, relies on an extension of the system with a dialog management infrastructure that allows requesters to know the access control policy they need to satisfy, and, thus, enables them to select a proper set of credentials to show [ACK<sup>+</sup>10]. Such communication infrastructure can be useful in another way: all the evaluations to indeterminate in those cases for which the server is missing information would be avoided, because the framework would introduce the possibility of notifying requesters which information is still required to give them the possibility to provide it and acquire access to the desired resource.

### 18.3.1 Policy Sanitisation

With the enrichment of access control systems with communication from servers to users on the enforced policies, another type of privacy issue rises.

For instance, suppose that an authorization imposes that attribute *nationality* should be equal to “US”. Should the server communicate such a condition to the requester? Or should it just inform the requester that it has to state her nationality? There is no unique answer to such question, and which one is to be preferred depends on the specific context we are focusing on.

However, communicating the complete policy (i.e., the fact that the policy will grant access if the nationality is US) favors the privacy of the requester. A requester can know, before releasing credentials or information to the server, whether the release will be sufficient to acquire access to the service. A client associated with a non-US user can avoid disclosing the nationality of the user. On the contrary, disclosing only a part of the policy protects the server’s privacy. Access control policies are considered sensitive information, and as such they need to be protected. For instance, while the server might not mind disclosing the fact that access to a service is restricted to US citizens, it might not want to disclose other conditions as they are considered sensitive.

Consider an authorization allowing access to a service to those users who work for an organization that does not appear in a Secret Black List (SBL) kept by the server. The corresponding condition, for some credential  $c$  carrying employment information, would then be:  $c.employer \notin SBL$ . Communicating the complete policy to the requester would imply releasing the condition above, together with the state of black list  $SBL$ . Also, assuming the context of  $SBL$  is not released, the requester will know, in case she is not granted access, that her employer is black listed. This is clearly an information the server does not wish to disclose; rather the server will want to maintain confidential the condition and simply state that the employment certificate is required.

Among the two extremes of the current XACML approach of simply returning indeterminate, on the one hand, and of completely disclosing the policy, on the other hand, there are other options offering different degrees of protection to the server policy and of information communicated to the requester.

Each condition appearing in the policy can then be subject to a different disclosure policy, regulating the way the presence of such a condition should be communicated to the requester. We can distinguish five different disclosure policies, and each one can be used independently in any condition appearing in an expression. We assume to have a condition  $c.A \pi v$  (where  $A$  is an attribute,  $\pi$  a predicate, and  $v$  a value) on a credential of type  $T$  issued by authority  $S$  ( $c :: T$  issued-by  $S$ ) to illustrate the effects of the disclosure policies, as below, where we include the portion of a condition not to be disclosed in square brackets.

- *None*. Nothing can be disclosed about the condition. It corresponds to the XACML approach of communicating that the outcome of the policy is indeterminate, since there are conditions that cannot be evaluated. Formally, the condition will appear in the policy on the server's side completely included in square brackets, that is,  $[c :: T \text{ issued-by } S]$  and  $[c.A \pi v]$ .
- *Credential*. Only the information that there is a request for a credential of a specific type can be disclosed. There is no further information on how such credential is to be evaluated. The condition will appear on the server's side like  $c :: T [ \text{ issued-by } S ]$  and  $[c.A \pi v]$ .
- *Attribute*. The information that an attribute needs to be evaluated can be released; no information can be released on the control that will be enforced on such an attribute:  $c :: T \text{ issued-by } S$  and  $c.A [\pi v]$ .
- *Predicate*. The predicate with which the attribute in the condition is evaluated can be released; no information can be released on the value/s against which the evaluation is performed. On the server's side the condition is written as follows:  $c :: T \text{ issued-by } S$  and  $c.A \pi [v]$ .
- *Condition*. The condition can be fully disclosed as it is. Formally, the condition will appear in the expression with no square brackets, signaling that no component is subject to disclosure restriction, i.e.,  $c :: T \text{ issued-by } S$  and  $c.A \pi v$ .

Table 18.1 summarizes the different disclosure policies reporting the formal notation with which they appear in the server's policy and the relevant communication to the client in the dialog. A similar approach is presented in [ADP<sup>+</sup>10].

Note that the disclosure policies of the server, affecting the information released to the requester about the conditions appearing in the policy, also impact the way the requester can satisfy the conditions. In particular, the *credential* policy implies that the requester will not know which information in the credential is needed and therefore will have to release the credential in its entirety. The *attribute* policy implies that the requester can selectively disclose the attribute in the credential. The same happens with the *predicate* policy, where the requester also knows against which predicate the attribute will be evaluated. Finally, in the case of the *condition* policy, the requester can either provide the attribute (but it can assess, before submitting,

Disclosure policy	Condition at server	Condition at client
<i>none</i>	$[c :: T \text{ issued-by } S]$ $[c.A \ \pi \ v]$	$[\ ]$ $[\ ]$
<i>credential</i>	$c :: T \text{ issued-by } S$ $c. [A \ \pi \ v]$	$c :: T \text{ issued-by } S$ $c. [\ ]$
<i>attribute</i>	$c :: T \text{ issued-by } S$ $c.A [\pi \ v]$	$c :: T \text{ issued-by } S$ $c.A [\ ]$
<i>predicate</i>	$c :: T \text{ issued-by } S$ $c.A \ \pi [v]$	$c :: T \text{ issued-by } S$ $c.A \ \pi [\ ]$
<i>condition</i>	$c :: T \text{ issued-by } S$ $c.A \ \pi \ v$	$c :: T \text{ issued-by } S$ $c.A \ \pi \ v$

Table 18.1: Disclosure policies and their effect on conditions.

whether such a release will satisfy the condition) or provide a proof that the attribute satisfies the condition.

Consider a policy stating that “a user can access a service if her nationality is Italian, her city of birth is Milan, and her year of birth is earlier than 1981”. Suppose that all attributes mentioned in the policy must be certified by an X.509 identity card released by IT.Gov. The policy is formally stated as:

- 01: own *id::IdentityCard* issued-by [IT.Gov]
- 02: where *id.method* = ‘X.509’  $\wedge$  *id.nationality* = [‘Italy’]  $\wedge$
- 03:     *id.city\_of\_birth* = ‘Milan’  $\wedge$  *id.year\_of\_birth* < [1981]

Here, the square brackets representing the disclosure policies implicitly state that: *i*) conditions on attributes *method* and *city\_of\_birth* can be disclosed as they are; *ii*) conditions on the issuer and attribute *nationality* need to be protected by hiding the control that will be enforced on them; and *iii*) condition on attribute *year\_of\_birth* needs to be protected by hiding the value against which the evaluation will be performed. If the above policy applies to a request submitted by a requester for which the server has no information, the following conditions are communicated to the requester.

- 01: own *id::IdentityCard* issued-by [ ]
- 02: where *id.method* = ‘X.509’  $\wedge$  *id.nationality* = [ ]  $\wedge$
- 03:     *id.city\_of\_birth* = ‘Milan’  $\wedge$  *id.year\_of\_birth* < [ ]

The requester can satisfy such conditions by releasing an identity card containing the requested attributes.

## 18.4 Integration into XACML

The goal of this section is to describe how one can bring privacy-preserving access control (as described above) to the real world [ADN<sup>+</sup>10] by leveraging the status of

XACML as the *de facto* standard in access control languages. To do so, a number of issues need to be addressed: (1) First, XACML does not manage attributes bundled in credentials, and thus does not allow for distinguishing whether two attributes are contained in the same credential or in different ones. (2) Second, XACML prescribes that the requester communicates all of her attributes to the server for the evaluation of the access control policy, which is problematic from a privacy perspective. Some technologies such as SAML, OpenID, and anonymous credentials, offer the possibility to reveal only a subset of the attributes contained in a credential. Such features can be exploited by first communicating the policy to the requester, so that she can disclose only the information necessary for the access. (3) Third, XACML and SAML merely allow requesters to reveal concrete attribute values, rather than allowing them to prove that certain conditions over the attributes hold. This further privacy-preserving feature can be obtained by leveraging the cryptographic power of anonymous credentials.

We envisage a setting where the requester owns a set of credentials obtained from various issuers, possibly implemented in different credential technologies. Servers host resources and protect them with policies expressed in an extended version of XACML. Users requesting access to a resource receive the relevant policy, which describes the requirements on the requester's credentials in order to be granted access. The policy may include *provisional actions*, i.e., actions that the requester needs to fulfill prior to being granted access. Subsequently, the requester inspects the policy and, if she has the necessary credentials to satisfy it, she creates a *claim* over a suitable subset of her credentials, which can describe (1) values of attributes contained in these credentials, (2) conditions over non-disclosed attributes, and (3) the fulfilled provisional actions. The requester derives (technology-specific) *evidence* for the claim to convince the server of its correctness and her ownership of the credentials. Afterwards, the requester makes a new request for the resource, but this time she includes the created claim and evidence. The server verifies the validity of the evidence with respect to the claim and evaluates whether the policy is fulfilled by the claim. Access is granted or denied accordingly.

Before describing the extensions that we made to XACML, we give a short introduction to it. XACML defines an XML-based access control policy language as well as a processing model for evaluating the policies on the basis of a given XACML access request. Such a request specifies by means of *attributes* which *subject* (i.e., who) wants to perform which *action* (i.e., do what) on which *resource* (i.e., on what).

An XACML policy is basically a structured set of *Rules* that define positive or negative authorisations (*Permit* or *Deny* rules). A rule contains a *Target* and a *Condition* that together determine its applicability, i.e., to which access requests it applies.

An XACML system consists at least of a policy enforcement point (PEP), a policy decision point (PDP), a policy administration point (PAP) and a context handler (cf. [Figure 18.1](#)). Access requesters issue their requests to the PEP who is responsible for enforcing the access control decisions that are rendered by the PDP on the basis of the request. The PDP makes decisions with respect to policies that are created and maintained by the PAP. The context handler is an intermediate compo-

nent between the PEP and the PDP that buffers the attributes that were given to the PEP in the request and provides them to the PDP on demand.

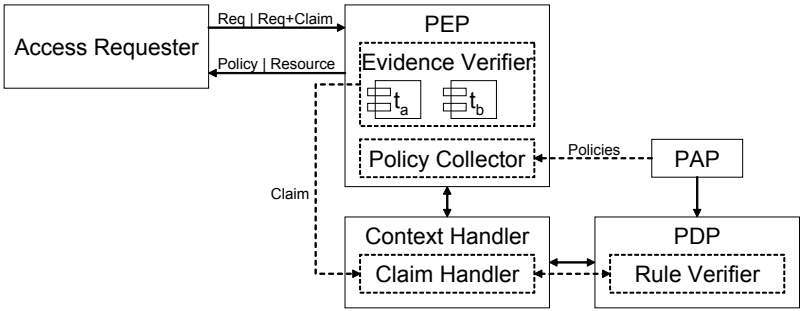


Fig. 18.1: XACML architecture with extensions. Standard XACML components are depicted with *solid* lines. Extensions are depicted with *dotted* lines.

### 18.4.1 Credential-Based XACML

The language extensions that we propose to XACML go beyond the standard extension points. All proposed extensions are in line with the semantics of existing XACML language constructs though, i.e., we do not alter the semantics of existing elements or attributes.

XACML rules that contain credential requirements can only have effect `Permit`. Rules with effect `Deny` are pointless as they essentially require that the requester *does not* have a certain credential. Assuming that the requester’s goal is to obtain access, she can always pretend not to have the specified credentials.

Our extensions enable policy authors to express conditions on the credentials that a requester must own and the actions that she must perform prior to be granted access. To this end, we augment the *xacml:Rule* element with optional *Credential-Requirements* and *ProvisionalActions* child elements. The former describes the credentials that the requester needs to own and the conditions these credentials have to satisfy. The latter describes the actions she has to perform.

#### Credential Requirements.

To express credential-based access control policies, the language needs a way to refer to the credentials that bundle several attributes together. Cross-credential conditions are another important use case: for example, the policy language must allow for expressing that the names on a credit card and on a passport must match, or that the expiration date of an entry visa is before the expiration date of a passport.

To this end, *CredentialRequirements* contains a *Credential* child element for each credential involved in the rule. The *Credential* can contain *AttributeMatchAnyOf* child elements that allow for comparing an attribute of that credential to a list of values. The *CredentialRequirements* also contain a *Condition* where conditions on the credentials' attributes can be expressed. Inside a condition, one can refer to an attribute `AttrId` within a particular credential by means of *CredAttributeDesignator*. An example rule is given in Figure 18.2.

```
<Rule Effect="Permit" RuleId="rule2">
  <xacml:Condition>
    <!-- XACML condition relevant for the rule's applicability -->
  </xacml:Condition>
  <CredentialRequirements>
    <Credential CredentialId="pp">
      <AttributeMatchAnyOf AttributeId="pl:CredentialType">
        <MatchValue MatchId="xacml:string-equal">un:PhotoID</MatchValue>
      </AttributeMatchAnyOf>
      <AttributeMatchAnyOf AttributeId="pl:Issuer">
        <MatchValue MatchId="xacml:anyURI-equal">http://www.usa.gov</MatchValue>
      </AttributeMatchAnyOf>
    </Credential>
  </CredentialRequirements>
  <Condition>
    <xacml:Apply FunctionId="xacml:date-less-than-or-equal">
      <CredentialAttributeDesignator CredId="pp" AttributeId="un:DateOfBirth"/>
      <xacml:Apply FunctionId="xacml:date-subtract-yearMonthDuration">
        <xacml:EnvironmentAttributeDesignator AttributeId="xacml:current-date"/>
        <xacml:AttributeValue DataType="xs:duration">P21Y</xacml:AttributeValue>
      </xacml:Apply>
    </xacml:Apply>
  </Condition>
</CredentialRequirements>
<ProvisionalActions>
  <ProvisionalAction ActionId="pl:Reveal">
    <xacml:AttributeValue DataType="xs:anyURI">un:Sex</xacml:AttributeValue>
    <xacml:AttributeValue DataType="xs:anyURI">pp</xacml:AttributeValue>
  </ProvisionalAction>
</ProvisionalActions>
</Rule>
```

Fig. 18.2: Example rule stating that access is granted to users who are at least twenty-one years old according to a piece of PhotoID issued by the US government, but only after revealing the gender mentioned on the same piece of PhotoID. Namespace prefix `xacml` refers to the XACML 3.0 namespace, `xs` to XML Schema, `pl` to <http://www.primelife.eu>, and `un` to <http://www.un.org>.

Conditions on credential attributes are expressed using the same schema as the *xacml:Condition* element (extended by *CredAttributeDesignator*), but are contained in a separate *Condition* child element of a *Credential* element.

### Provisional Actions.

The *ProvisionalActions* element contains the actions that have to be performed by a requester prior to being granted access. The types of actions that we model are

the revealing attributes (optionally to third parties and optionally with an attached data handling policy) and the signing of statements (to express consent). Each provisional action is contained in a *ProvisionalAction* element that includes an action identifier as an attribute `ActionId`. For example, the policy in [Figure 18.2](#) contains the requirement to reveal the gender as specified on the identity card.

### 18.4.2 SAML as Claims Language

Here we describe how we extend SAML for transporting the claims as defined in the beginning of this Section. SAML is a standard allowing for the exchange of certified attributes bundled together into *assertions*, which are similarly structured as credentials. The standard, however, only allows for the exchange of attribute values but not conditions on such values nor notifications of provisional action fulfilment. To address these issues, we use the standard's extension points to embed our *Condition* and *ProvisionalAction* elements into SAML assertions.

### 18.4.3 XACML Architecture Extensions

In the following, we sketch how we adapt the XACML architecture such that (1) the credential-based XACML policy applicable to a request is communicated to the requester, and (2) the policy can be evaluated on the basis of the provided SAML claim. The modified architecture maintains all standard XACML functionality, i.e., the modifications are *extensions* that do not substitute existing functionality and that are usable in combination with standard features.

We adapt the XACML communication model for allowing the following two-round pattern. In the first round, the requester specifies a resource and obtains the relevant policy from the PEP; in the second round, the requester sends the same request with an additional SAML claim. Resending the request is necessary because the XACML architecture is stateless, meaning that the individual components do not maintain information across multiple rounds. A PEP's response in the first round is embedded in an *XACMLPolicy Assertion* element (cf. SAML profile of XACML [OAS05b]), to which the requester is supposed to reply with an appropriate SAML claim. The PEP grants or denies access depending on the claim's validity and the decision of the PDP.

We need to modify the PEP such that it obtains all policies applicable to a user's request and then sends them in a pre-evaluated version to the user. The pre-evaluation substitutes known attributes, e.g., environment attributes such as time and date, with concrete values.

When the PEP receives a request with an attached SAML claim, it has to verify the validity of the claim and make it available to the PDP. To verify the validity of the claim evidence, we extend the PEP with an *evidence verifier* component (cf. [Figure 18.1](#)). For every supported credential technology  $t$ , this component has a plug-in

that can verify evidence specific to this technology. To make the claim available to the PDP, we introduce a *claim handler* component within XACML's context handler. If the claim is valid, the PEP forwards it to the claim handler, which buffers it so that it can be retrieved by the PDP. The PEP then forwards the request (without attached claim) to the PDP.

A PDP evaluates a request from the PEP as usual with respect to the rules in the policy. However, rules with credential requirements or provisional actions are treated specially. For such rules to yield a `Permit` decision, not only its applicability (specified by its *target* and *condition*) is relevant, but also the fulfillment of the credential-requirements and provisional actions, if any are specified. If so, the PDP fetches the claim from the claim handler. We extend the PDP with a *rule verifier* component that, for given credential requirements, given provisional actions, and a given claim, decides whether the claim implies the requirements and fulfills all the provisional actions. If so, then the rule evaluates to `Permit`, otherwise it evaluates to `Indeterminate`.

## 18.5 Concluding Remarks

By means of the research work illustrated in this chapter, we aimed at advancing access control technology with a specific focus and a wide-ranging impact: enabling all involved parties to request, evaluate, grant, and obtain access to services and data in a way that fulfills their privacy preferences in the best possible way. For the current state of the art to progress, many requirements were met, regarding both policy languages in general, and access control in particular.

Our main focus on anonymous credentials has obviously played a fundamental role in fully achieving the goals on data minimisation and anonymous or pseudonymous access control, brought even further by the introduction of sanitisation techniques to meet servers' privacy issues.

Our credential model, comprised of a list of signed attributes, provides a solid support to both role-based and attribute-based access control. Moreover, we have elaborated a declarative language that allows for the expression of high-level, compact policies that are easier to understand than the proposals made so far in the field. The policy model is indeed technology-independent, which greatly widens its application scope.

The simple structure of each credential paves the way for quick yet effective construction of ontologies of credential types and attributes, which not only allow for even more compact policies, but also support delegation mechanisms through sequences of credentials embodying chains of trust between certification authorities.

Finally, we embedded our proposals into an existing standard like XACML, whose policy prioritization and combination mechanisms are thus made available to policy editors.





# Chapter 19

## Legal Policy Mechanisms

Leif-Erik Holtz and Jan Schallaböck

**Abstract** Transparency is one of the core principles of data protection legislation in Europe, beyond Europe and all around the world. The European understanding is different than the American one as the European understanding is that individuals should be aware of ‘who knows what about them.’ Often enough the establishment of the European understanding is hard to enact, enforce and above all make understandable to the user because the user is confronted with a multitude of different purposes for data handling, often hidden in lengthy legal text of privacy notices especially when surfing the web. Therefore, a number of approaches are currently trying to tackle this problem, by offering the user tools and mechanisms for a better understanding of what is happening with their data. The work presented in this chapter is an outcome of PrimeLife’s research on Next Generation Policies, it aims at a better understanding of the legal aspects of the processing of personal data, by looking at the current status of this processing in different contexts and structuring these.

### 19.1 Introduction

The research on legal policy mechanism aims at developing the basis for taxonomies and partonomies that can be used as a starting point in defining vocabulary for technical and legal privacy policies and languages expressing the latter. What is the purpose for this research approach? Why are taxonomies and partonomies important means for defining vocabulary for privacy policies?

As described above, transparency is one of the core principles of data protection legislation in Europe [Com95], beyond [APE05] and all around the world [OEC80]. The common understanding is, that individuals should be aware of ‘who knows what about them’ [Cou83]. This concept is supported by the principle of purpose or collection limitation. This principle stipulates, that any collected personal information may only be processed for those purposes it was collected for. Roots of this principle

can be seen in what Nissenbaum calls ‘contextual integrity’ [Nis04, Nis10, Nis98], or equally in the sociological concept of ‘functional differentiation’ [Luh77]. They appear to be basic conditions of just communication and social interaction in democratic societies. Often enough, these principles are hard to enact, enforce and above all hard to understand for a user. The user is confronted with a multitude of different purposes, often hidden in the lengthy legal text of privacy notices’ especially when surfing the web.

A number of approaches are currently trying to tackle this problem, by offering the user tools and mechanisms for a better understanding of what is happening with their data (see below in the references). However, in most if not all of these approaches’ it is unclear what is actually necessary to communicate. The problem relates to the above. For a higher level of transparency, the user should be made aware of what actually happens to the data, who is processing it, and – if collected without the informed consent of the user, what data is processed (e.g., Cookies, IP-Addresses, clickstreams etc). While the latter elements may be easy to communicate (but still might need some further thought), the question of how to express in a simple way, how the data are processed, and for what purpose(s) they are collected, poses difficulties. The multitude of applications and uses of personal data are highly unstructured, as no comprehensive ontology exists, and no abstractions are apparent.

For developing a typology of the processing types of personal data an empirical approach, looking at the current practice, seems appropriate.

The work in PrimeLife’s Activity 5 on Next Generation Policies (cf. Part IV of this book), especially in the area of legal policy mechanisms, aims at a better understanding of the legal aspects of the processing of personal data, by looking at the current status of this processing in different contexts and structuring these.

## 19.2 Legal Framework for Processing Personal Data

Recent and ongoing work in user transparency and legal privacy policies is done in the area of Human-Computer-Interfaces (HCI), as well as in technical representations and functional descriptions of policies for privacy and data protection. The Article 29 Working Party has endorsed the use of multi-layered legal policies for websites [Par04], which has been implemented in several places on the web [Bro08]. Others have been developing tools and interfaces to support the user [Con07a]. Finally there are proposals to use iconography to simplify the recognition of a legal privacy policy for the user [Run06] [Fis06] that have gained some momentum [Pri09c].

For a formal description of privacy policies, P3P is an available specification [W3C01], and offers some structural reference. XACML [OAS08], EPAL [W3C03], Liberty’s Internet Governance Framework [Pro07], and WS-Policy [W3C06b] are discussed for further expressing rules for the processing of personal data. However, all of these specifications are very limited, or even completely lacking in offering

conventions on describing the aspects most relevant to the user: What is actually going to happen to the data, what purposes are they used for, under what conditions and with what obligations?

Last but not least the PRIME-Project [Con07b] did initiate some research for a more thorough ontology in this area and had reached some first results, which further research should take into account, as well.

General legal requirements for processing personal data are defined in the Data Protection Directive 95/46/EC (DPD) and the Directive on Privacy and Electronic Communications 2002/58/EC (DPEC). DPD and DPEC mandate member states of the EU to implement its normative content into their respective jurisdictions. The state of the implementation is documented by the EDPS (European Data Protection Supervisor). Legal requirements for privacy policies encompass data collection as well as data handling, i.e. the processing. The core element of the DPD is only allowing the collection and processing of personal data, if there is a legal basis or if the data subject unambiguously gave his consent, cf. Article 7 of the DPD.

In any case of data processing, it is always worthwhile to review the six legal grounds for data processing to ensure that one of them is present and that the data are therefore processed legitimately [Kun07].

The legal grounds are the following:

- The data subject has unambiguously given his consent (see Article 7 (a) (all following Articles refer to the DPD)); or
- Processing is necessary
  - For the performance of a contract of which the data subject is party, or in order to take steps at the request of the data subject prior to entering into a contract (see Article 7 (b)); or
  - For compliance with the legal obligations to which the controller is subject (see Article 7 (c)); or
  - In order to protect the vital interests of the data subject (see Article 7 (d)); or
  - For the performance of a task carried out in the public interest or in the exercise of official authority vested in a controller or in a third party to whom the data are disclosed (see Article 7 (e)).

If the collection is based on the latter, the following information has to be provided to the data subject (amongst others):

- The data controller has to be declared as well as
- The types of data collected and
- Legitimate purpose of the processing has to be defined, cf. Article 6 and 7 of the DPD.

In the further process, the data controller has to ensure that the data is kept accurate and only used for the purposes declared *ex ante*. While the former are relatively easy to define, a specific problem of the legal requirements is the defining of the purpose of data collecting.

The difficulty of how precise the policy language has to be, can also be seen in light of the DPD. According to Article 12, lit a, bullet point 3, the data subject has the

right of information about the data handling. This is also the expression of the right of informational self-determination of the data subject. Therefore it is necessary, that the data collector displays at least applied information about data handling in his policy. And the data collector also needs a precise policy to comply with his promises when processing the data internally in his own business. To ensure the implementation of legal requirements during data handling, the data collector has to advise his employees on how to process the collected data. To avoid mistakes in handling data, the advice has to be as precise as possible.

## 19.3 Gaps in Current Policy Language Approaches

Sticky policies promise to improve the state of the art in data protection, both on the level of better control, and on the level of increasing transparency. A sticky policy is usually the result of an automated matching procedure between the data subject's data handling preferences and the data controller's data handling policy. Associated to a resource, it is the agreed-upon sets of granted authorisations and promised obligations with respect to a resource. Sticky policies are policies that control how data is to be accessed and used and that accompany data throughout an entire distributed system [CL08].

A privacy policy that would support privacy by way of sticky policies would need to implement legal requirements for the employees of the data collector as well as for the data subject. The maxim of transparency in data collecting and data handling also argues for the assumption that a privacy policy has to be as precise as possible. This clarifies the need for a precise and well-implemented privacy policy. A number of different and "complementary" approaches are currently being taken to support legal compliance in current IT-Systems.

The goal of these approaches is to support compliant use of the system by mixing technological and organisational mechanisms. The legal framework is ideally already introduced when defining the specification of the system. Numerous policy languages currently available or under development address this. In the following a two approaches, XACML (extended) and P3P will be highlighted and analysed with respect to their potential to support legal compliance.

### 19.3.1 XACML

Extensible access control markup language (XACML) is an XML-based language for expressing and interchanging access control policies [Pri09b, page 48]. The language's core functionalities are geared towards access control, but it also offers standard extension points for defining new functions, data types, policy combination logic and more. In addition to the language, XACML defines both an architecture for the evaluation of policies and a communication protocol for message interchange,

as well as supporting multiple subject specifications in a single policy, multi-valued attributes, conditions on metadata of the resources and policy indexing. XACML is an applicable mechanism for technically implementing legal requirements of access control. The control of data handling on the other hand is not fully covered by XACML, but may potentially be achieved by means of its extensibility. The benefits of XACML on the one hand and the shortcomings on the other hand have already been depicted in detail in Chapter 20.

During the development of PrimeLife, some effort has been spent on extending XACML. This work suggests a new obligation handling mechanism, taking into account temporal constraints, preobligations, conditional obligations and repeating obligations together with a down-stream usage authorisation system, defining the access control rules under which personal information collected by an entity can be forwarded to a third party [ABD<sup>+</sup>10].

Part of this work is based on the concept of trusted credentials. XACML was extended with data handling and credential capabilities. The overall structure of XACML was maintained, and a number of new elements to support the advanced features the language offers were introduced. It can be used by the data controller as well as by the data subject. The result of the completed research towards XACML-PrimeLife is that it is a suitable way to display the data collecting party, but that there is still difficulty in displaying the purpose of the data processing. Currently, the research forms a wrapper in this regard that can be used to contain elements of the Platform for Privacy Preference (P3P), but also could include different ontologies.

### **19.3.2 P3P**

For a formal description of privacy policies, P3P is an available specification that offers some structural reference [W3C01]. P3P enables websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents.

Concern has been raised that P3P may be too complicated to be understandable for users. However, P3P has the advantage of describing the aspects most relevant to the user: what is actually going to happen to the data, what purposes they are used for and under which conditions and with what obligations [Pri09a, page 28].

Moreover, the expressiveness of P3P, especially in regard to describing purposes, is limited (albeit extensible). The predefined set of purposes was limited to so-called secondary purposes in the first specification (<http://www.w3.org/TR/P3P/>), which was then complemented with a flat partonomy of some twenty “primary purposes” in Version 1.1 [W3C01]. Taking into account the requirements for displaying the purpose to the user, as well as requirements of internally achieving legal processing of the data within the limits of the purpose, this appears to be too limited, as we have shown in a number of scenarios (below, Section 19.5) and will need further extension in its vocabulary, potentially by way of a more comprehensive ontology, or at least a taxonomy or partonomy.

## 19.4 Methodology

To derive a basis for developing a vocabulary following the criteria specified for sticky policies, empirical approaches were analysed and evaluated with regards to their value for developing such a basis:

- The German corpus juris, especially those norms in federal law, allowing the processing of personal data by public bodies and agencies,
- “Verfahrensverzeichnisse” a regulation specific to German data protection law: it mandates data controllers to maintain a list of those processes, within which they are processing personal data, cf. 4 lit. e BDSG,
- Privacy Statements from the Internet, possibly in cooperation with further entities,
- PrimeLife Use Cases.

The goals of such an approach are to find and to define vocabularies for the following attributes:

- Processes and services for personal data,
- Purposes of Processes and a partonomy/taxonomy thereof, to be sorted by relevance,
- Typical sets of data, expressed as a partonomy of data types and data sets,
- Data types and possibly qualifications/attributes (such as sensitive information as defined by 95/46/EC),
- Reference to the legal basis for the processing (norms, legal privacy policies),
- Text elements from legal privacy policies,
- Possible further elements, especially obligations, such as logging, deletion, blocking, further information (e.g., in case of incidents), retention periods (currently not included in the research),
- Categories of data processors, and a partonomy/taxonomy thereof (currently not included in the research).

It is due to the very nature of data processing that it is impossible to conduct a comprehensive ontology towards this topic, but there seems to be the option of an advanced taxonomy and possibly an ontology, which would cover processing to a certain, possibly defined, level of detail and brevity.

### 19.4.1 Looking into Privacy Policies

Our research, conducted on a limited number of privacy policies, has shown differing results on the usability of the analysed policies. We have analysed 34 privacy policies, most of them from dutch websites and therefore most of them under the legislation of the DPD. The following results of the analysis can be measured: Some of the privacy policies implemented the legal requirements for collecting data in a

legally compliant manner while others did not. The implementation of legal requirements for data handling on the other hand was not displayed very well. The main problem in privacy policies seems to be the description of data handling. This also is a legal requirement, but only few policies implemented it. More detailed information about the analysis can be found in PrimeLife Heartbeat H5.2.2.

Many of the analysed policies did not have the level of transparency that they should have provided, which raised some concern, as to whether they complied with the requirements of the legal framework in place. This alone would certainly have been an interesting field for further research, but was not within scope the conducted research.

Another difficulty that arises when analysing such policies is a possible bias of interpretation. One approach to balance this bias, would have been to have each policy analysed by two researchers, with a third looking at those policies, where the previous results differed. This approach, although promising, was dismissed, taking into account the resources available for the research, but it should be taken into consideration for further research.

### ***19.4.2 Looking at the Law***

The German corpus iuris appeared to be another interesting empirical basis. Due to the specific construction within German law, the processing of personal data by governmental agencies needs a specific legal basis, therefore a broad data set was to be expected.

After a selection of laws were analysed, evaluation hinted towards the fact, that this approach might not be the most effective. The specific language chosen in many cases would remained on too high of a level to yield results of the granularity necessary. Again, this indicates, that even lawmakers do not achieve a reasonable level of transparency in their laws, which unfortunately also makes this approach inefficient.

A similar, but slightly different approach based on German law, was using German *Verfahrensverzeichnisse* (i.e. literally: processing directories). This specificity of German law, mandates data controllers to describe each process wherein personal data is processed. While the former approach seemed promising, the available material from the German *Verfahrensverzeichnisse* was too limited to come to an effective result. Although controllers are obliged to have these directories, they are often not kept up to date. It was expected, that organisations would react with an outcry, if asked to provide their *Verfahrensverzeichnisse*, especially when asked by a data protection authority, which is the partner employing the researchers conducting the research. Even though, due to these implications, it was refrained from following this approach, it may have triggered further action within the agency, promising follow this approach at a later point in time.

Further research concentrated on researching selected use cases. On the one hand, working with use cases has the disadvantage that a very comprehensive ontology does not seem to be in reach on this basis. On the other hand, there are several



advantages, in working with use cases. For example, each possible scenario could be analysed quite comprehensively and the problems in real life in connection with privacy protection can be displayed very well.

## 19.5 Use Cases

During the course of PrimeLife, the project developed a set of use cases. Thus there was already a strong foundation to build upon. Therefore the approach promised a high level of compatability for other research aspects within the project. One of the use cases that was analysed and will be analysed further is the scenario of a 'classical' online-shop (which had already been a good scenario to display the problems concerning privacy protection during the PRIME project). Subsequently, the methodology was also performed on a social network site's use case. Social network sites comprise a higher degree of complexity related to purposes of data handling because many different constellations may appear and many different data controllers or processors respectively may exist. Given the complexity of social network sites, we were also able to assess the limits of the methodology.

### 19.5.1 *Online Shopping*

For reasons of brevity, the use cases can only be described by example. For this chapter, age verification was chosen, as it has a number of interesting implications. A more comprehensive overview can be depicted from the figure below, cf. Fig. 19.1.

Coming back to the example: One of many questions for a web shop scenario is the task of age verification. If a shop - such as Amazon - sells comics like Donald Duck, there is no need to verify the age of the user. The comic has no youth endangering contents and is therefore free of age limitations.

A typical case of national laws allowing minors to engage smaller contractions, is the German 'Taschengeldparagraph' 110 BGB, allowing minors to conclude valid contracts with their pocket money. Here, there is no need – and no legal basis – for collecting the data concerning the date of birth. If the same shop sells alcohol – such as wine – there is an age limitation and the legal basis for collecting the data concerning the date of birth is the necessary to comply with the legal obligation of not selling alcohol to minors. Therefore, the privacy policy has to display the purpose for collecting a personal data such as the date of birth. Another question according the web shop is, which data is needed for payment. This depends on many parameters. One conclusion of the finished research and the estimated results is, that the research on concrete scenarios will lead to an available result. The past and the further research on the concrete scenario of a online shop might approximate a number of problems concerning data protection.

After a thorough analysis it was clear, that a simple scenario such as the one on ‘Online Shopping’ can yield a plentitude of different data handling purposes and obligations. For a better understanding of the different purposes, we tried to define a selection of them inter alia with their possible legal basis.

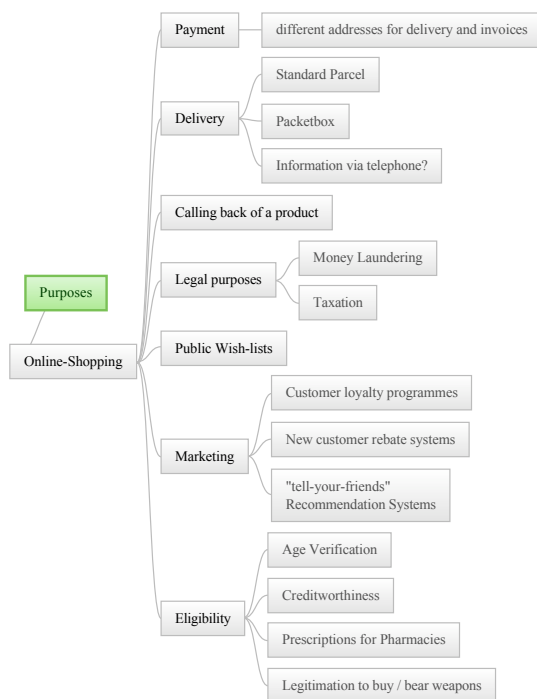


Fig. 19.1: Overview of different purposes in an Online-shopping scenario. The figure shows the multitude and hierarchy of purposes. The latter, however maybe misleading, often a definite hierarchy of purposes cannot be described.

With regards to some of the scenarios, the collection may also already be legal simply because it is "necessary for the purposes of the legitimate interests" of the shop (cf. Art. 7, lit. e of the DPD), but in many other scenarios the controller will rely on the user actively giving consent to the processing of the information. The complete range of all purposes for collecting data have to be displayed, to broadly inform the user. This would make an informed decision impossible. On the other hand, broad information is the assumption for active consent of the user towards collecting his personal data. Therefore the purposes of all scenarios of the use case 'online shop' have to be displayed in a way that the user can understand. The analysis of the web shop use case illustrated the fact that purposes of data handling can be visualized quite well in this use case and be made specific on a very granular level.

From this perspective the web shop use case appears to be a very well-defined case. Purposes can be differentiated clearly and depicted relatively easily.

### ***19.5.2 Social Networking***

Social Networking implies some very specific legal questions, many of which are currently under heavy discussion within jurisprudence. They may therefore currently very well be considered as one of the more difficult legal challenges in privacy legislation. It therefore seemed reasonable to believe, that if a model could be developed for Social Networking, it would be possible to cover other areas with a similar approach. Over and above, better understanding privacy in Social Networks is one of the core research areas for PrimeLife, making studying a social network use case a logical step.

Purposes in social network sites can vary greatly, as will be shown. The question was: how can purposes in social network sites possibly be defined? According to the definitions of social network sites, this depends on different factors. One of the factors is the differentiation between those purposes defined by the provider of the service and those defined by the user.

Provider-defined purposes in social network sites can be quite complex. The purposes for data handling depend inter alia on the different purposes of the social network sites per se. For example, in those social network sites that involve primarily commercial or business objectives, other purposes may occur than in social network sites addressing leisure and recreational activities. We can therefore conclude that business oriented networks tend to define purposes for data handling mainly to promote the career benefits for the users. Other social network sites have different purposes such as “getting in touch with old classmates or friends” and “getting concrete answers to questions inside the network” [Lin10]. Such a network that is purely career oriented obviously defines other purposes for data handling. Apart from the network’s business model and niche, many different functionalities can be used within social networks. For example, Facebook offers functionalities such as mobile phone usage or sms usage [Fac10]. The different applications again have different purposes and therefore different purposes for data handling. Thus, social network sites offer a (potentially unlimited) wide variety of user defined purposes for status messages and thereby for data handling.

Examples for provider side defined use cases are therefore:

- Status messages, including status messages in closed user groups and public status messages,
- Designing own profiles,
- Personal messaging,
- Social search,
- Uploading Photos, including Tagging and commenting and Location data.

On the other hand there are user-defined purposes, which can differ largely from provider-defined purposes. Users are self-defining the purposes for data handling on a case by case basis, which makes it very difficult to assess on a system side *ex ante*. These user-defined purposes in turn also tend to be more defined by aspects of usage culture between users. In this area the research was therefore following a slightly different approach, which eventually was integrated into a separate project: One way of predefining purposes in a user-defined scenario might be the privicon approach [CSBS], where users can predefine how the recipient of an email should handle the mail.

## 19.6 Results and Further Research

The research completed and described within this chapter leads to the conclusion that many data controllers act on the assumption that it is precise enough to display the legitimate reason of the data collector on handling data as a legal basis (see German Federal Data Protection Law (BDSG) [Bun08], article 28, deposit 1, number 2 or article 7, letter d of the DPD) as part of their policies. This ‘catchall element’ – legitimate reason – is used as a general reason (BDSG, article 28, marginal number 1). However, this should not be the state of the art of privacy policies, as it does not allow a reasonable level of transparency for the data subjects.

Our research has shown that to avoid this for the future, a more precise description of policies is necessary. Moreover, it has also demonstrated that this can be done, at least in the selected cases. The completed work includes and compares different methodological approaches, the research based on use cases as well as the research with larger empirical bases. All approaches had a common goal in mind: the need of having access control policy languages that, on one hand, provide access control functionality and, on the other hand, protect the privacy of the involved parties and of their personal information.

After carefully analysing broader empirical approaches, a promising and extensive analysis, yielded extensive efforts. To be more precise: the empirical analysis of German data protection law alone, and privacy policies of German sites only, would be immense. An analysis of published privacy policies yielded similar limitations. Especially when trying to effectively rule out bias of interpretation, the effort doubles or triples. The completed empirical research also leads to the conclusion that all analysed approaches have deficiencies, as was shown above. This is why a mixed approach is taken: empirical research is supplemented by use case-based analyses. This approach although it did not promise the comprehensiveness of an empirical analysis, but rather focused on an exemplification of the required expressivity of the language. It includes, however, the advantage, of being able to look deeper into the technical processes underlying the respective policies, rather than solely looking at what is published in privacy policies. One of the benefits of this approach is the fact that it is easier to handle a known system as a basis of research than to handle unknown systems.

Nevertheless, future research will have to take all possible options into account, if a more comprehensive overview of data handling practices is to be achieved. A draft database structure for collecting such material was also developed as part of the research. The use case scenario ‘online shopping’ has also suggested that there are certain shortcomings in displaying data processes via the specifications that P3P offers, or at least that a higher expressivity is not only doable, but also sensible. However, future research will have to look into a careful comparison of both approaches by matching the results of the use case described herein to the expressivity of P3P.

More detailed information about the ongoing work and results achieved can be found in PrimeLife Deliverable D5.2.3 [Pri11b], and in PrimeLife Heartbeat H5.2.2 [Pri11a].

## Chapter 20

# Policy Implementation in XACML

Slim Trabelsi and Akram Njeh

**Abstract** This chapter presents the implementation details of the PrimeLife policy engine (called PPL engine). This engine is primarily in charge of interpreting the policies and the preferences defined by the Data Controllers and the Data Subjects. Additionally, this engine is responsible for the enforcement of the privacy rules specified by the user. The enforcement is characterised by the application of the access control rules, the execution of the obligations and the generation/verification of the cryptographic proof related to the credentials. In this chapter we describe the architecture of this engine, the structure of policy language, and finally the data model of the implementation.

### 20.1 Introduction

Since the PPL language is specified as an extension of the XACML (eXtensible Access Control Markup Language) language, the PPL engine is designed to run together with any XACML engine (that only handles XACML access control rules). The architecture chosen for the deployment of the PPL engine is symmetric because Data Subjects and Data Controllers have similar requirements: Deciding whether a given piece of personal information (resp. collected data) can be shared with a Data Controller (resp. Third Party); handling obligations associated with data; storing data and associated preferences (resp. sticky policies). Using the same architecture everywhere to handle scenarios where one party can have multiple roles (e.g., collecting data and then disclosing it to third parties). The PPL engine executes multiple tasks in automated ways such as: Enforcing access control policies, generating and verifying cryptographic proofs related to credential requirements, matching between data handling preferences and data handling policies, generating and enforcing sticky policies, checking Authorisation, controlling the downstream usage of data, handling obligations, etc.

## 20.2 Architecture

In this section, we will present the design phase of the PPL engine. We present the architecture of the PPL system by defining a high level and a detailed architecture.

### 20.2.1 High Level Architecture

As presented in (Figure 20.1) below, the high level architecture presents an abstract overview of the PPL architecture and the interaction between the different entities; Data Subject, Data Controller and Third Party (Downstream usage).

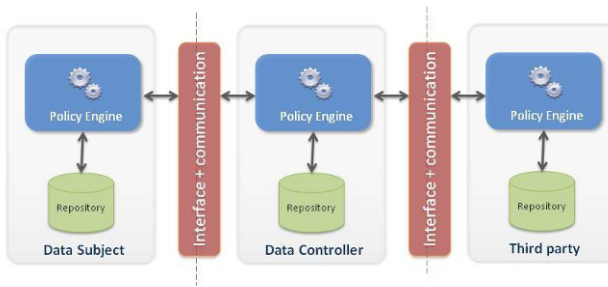


Fig. 20.1: High level architecture.

#### 20.2.1.1 Data Subject

*Policy engine:* This component is in charge of parsing and interpreting the privacy preferences of the Data Subject. This policy engine supports the entire PrimeLife Language capabilities (Preferences, Access control, DHP, Obligations, Credentials etc). For this reason, this module is replicated on the Data Controller side and the third party side.

*Repository:* Represents the PII and policy repositories. It is a database containing data owned by the Data Subject. This data could be composed of personal data, credentials, certificates, and other information that should be used during the interaction with the Data Controller application. It also contains the policy files representing his privacy preferences.

*Interface and communication:* This interface represents a communication interface with the Data Controller implementing the message exchange protocol.

### 20.2.1.2 Data Controller

*Policy engine:* This component is the same as the one described in the Data Subject section.

*Repository:* This repository represents a database that contains all the information collected from the Data Subject during her interaction with the Data Controller. This data represents PII's, credentials, certificates, and other information provided by the user. Also, this database contains the privacy policies related to the different resources and services that the Data Controller holds.

*Interface and communication:* This interface represents a communication interface with the Data Subject implementing the message exchange protocol. This interface plays the role of user interface described in the data subject section, in case of downstream interaction between the Data Controller and a Third Party.

### 20.2.1.3 Third Party

All the components supported by these actors are the same as those described in the Data Controller section. This is due to the fact that the Third Party plays the role of a Data Controller in case of downstream usage of the data.

## 20.2.2 Detailed Architecture

The entire architecture can be represented by three layers: The first one presents the user interface layer. The second, business layer, represents the core of the PPL Engine. The last layer represents the persistence layer that is in charge of data persistence.

### 20.2.2.1 Presentation Layer

The presentation layer is responsible for the display to the end user. The presentation layer contains two components:

- The policy editor: Displays and provides a way to manage all the information related to the Data Subject, Data Controller and the Third Party. This information can be the personal data (PII's, the credentials, etc), the privacy policy or preference, or the information involved during a transaction between the different entities. This component is not yet integrated to the current version of the demonstrator but should be part of the next release.
- The matching handler: Displays to the user the result of the matching. In the case of a mismatch, a set of tools are provided that allows the data subject to manage, or make an informed decision about, the mismatch.



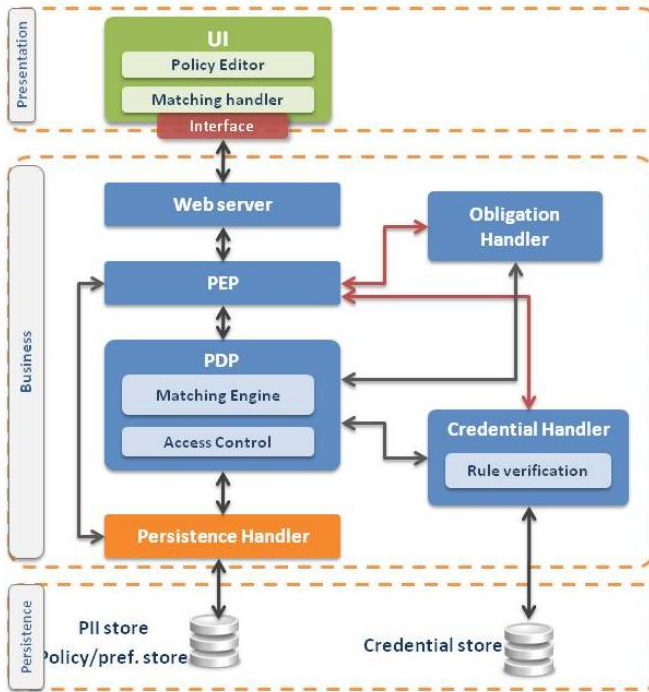


Fig. 20.2: Detailed architecture.

The UI layer should be independent from the business layer. For that, an interface component is deployed between these two layers in order to provide an abstraction level.

#### 20.2.2.2 Business Layer

The business layer, which represents the core of the PPL engine, is composed of four main elements that implements the new concepts introduced within PrimeLife. These components are:

- **Policy Enforcement Point (PEP):** This component formats and then dispatches the data to the corresponding component according to the state of the execution process. The decision made by the PDP is enforced in the PEP, meaning that if the PDP decided to provide data or enforce the access of one resource, this data or resource is collected, formatted and sent to the receiver through the PEP.
- **Policy Decision Point (PDP):** This is the core of the PPL engine where all decisions are made. It can be broken down into two subcomponents:

- *Matching engine*: This functionality matches between the preferences of the Data Subject and the privacy policy of the Data Controller. The matching is done to verify that the intentions of the Data Controller in terms of PII usage are compliant with the Data Subject's preferences.
- *Access control engine*: This component is in charge of the application of the access control rules related to the local resources. It analyses the resource query, checks the access control policy of the requested resource and decides whether or not the requester satisfies the rules.
- **Credential handler**: One of the new features introduced in PPL is the support of the credential based access control. This feature is implemented by the credential handler who manages the collection of credentials held by an entity, selects the appropriate credentials in order to generate a cryptographic proof and verifies the cryptographic proofs of the claims received from external entities. The credential handler component contains the subcomponent Rule Verification; the PPL policy contains a description of the credential requirements (for access control), the Rule Verification component evaluates whether the claim provided by a user that wants to access a resource satisfies the credential based access control rule.
- **Obligation handler**: This is responsible for handling the obligations that should be satisfied by the Data Controller/Third Party. This engine executes two main tasks; it sets up the triggers related to the actions required by the privacy preferences of the Data Subject, and executes the actions specified by the Data Subject whenever it is required.

The other components of the architecture play a secondary role in the concept introduced by the PPL engine:

- **Web server**: An embedded web server that represents the entry point of the core of the PPL Engine. It can be seen as an interface to the PEP.
- **Persistence handler**: Can be described as an interface between the business layer and persistence layer. It encapsulates access to the storage medium business objects. It makes transparent to the business layer location and storage model of the data it manipulates. In general, this layer is supported by a Persistence Framework. The defined objects in this layer are generally DAOs (Data Access Object). The persistence handler provides management functions to handle the DAOs known as CRUD (Create, Retrieve, Update, Delete) methods. The persistence handler provides the functions to manage the PIIs and the policies in the different databases.

### 20.2.2.3 Persistence layer

This layer is in charge of storing and persisting all the data involved in the system (PIIs, policies, preferences...). This layer is composed of two elements:

- **PII/Policy store**: This database (or other way of persistence, such as LDAP, etc) contains all of the information related to the PIIs and their related policies.

- **Credential store:** This database contains all the credentials and certified information held by an entity. Access to this store is exclusively dedicated to the credential handler component.

## 20.3 PPL Policy Language Structure

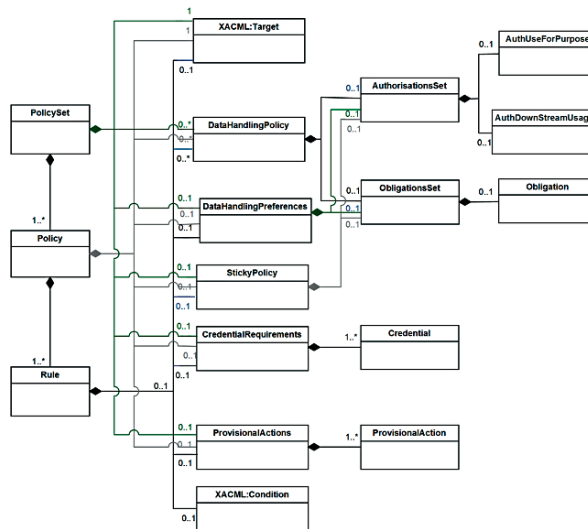


Fig. 20.3: Policy Language Structure.

The PPL language extends XACML 2.0 with a number of privacy-enhancing and credential based features. The PPL language is intended to be used:

- By the Data Controller to specify the access restrictions to the resources that she offers;
- By the Data Subject to specify access restrictions to her personal information, and how she wants her information to be treated by the Data Controller afterwards;
- By the Data Controller to specify how “implicitly” collected personal information (i.e., information that is revealed by the mere act of communicating, such as IP address, connection time, etc.) will be treated;
- By the Data Subject to specify how she wants this implicit information to be treated.

For that, we maintain the overall structure of the XACML language and we introduce a number of new elements to support the advanced features that our language aims to offer.

### 20.3.1 *PolicySets, Policy and Rules*

As in XACML, the main elements of our language are *PolicySet*, *Policy* and *Rules*. These elements are then inherited from XACML. Each *Rule* element has an effect, either *Permit* or *Deny*, that indicates the consequence when all conditions that were stated in the *Rule* have been satisfied. *Rules* are grouped together in *Policy*. When a *Policy* is evaluated, the rule combining algorithm<sup>1</sup> of the policy (as stated in an XML attribute of the *Policy*) defines how the effects of the applicable rules are combined to determine the effect of the *Policy*. *Policies*, in their turn, are grouped together in *PolicySet*; the effect of a *PolicySet* is determined by the effects of the contained *Policies* and the stated policy combining algorithm. Finally, different *PolicySet* can be further grouped together in parent *PolicySets*.

The *PolicySet*, *Policy* and *Rule* elements are composed by different elements:

- A *Target* (plain XACML:Target), which describes the *Resource*, the *Subject*, and the environment variables for which this *PolicySet*, *Policy* or *Rule* are applicable;
- *CredentialRequirements*, describing the credentials that need to be presented in order to be granted access to the resource; this element is not defined in the original XACML language;
- *ProvisionalActions*, describing which actions (e.g., revealing attributes or signing statements) have to be performed by the requester in order to be granted access; this element is not defined in the original XACML language;
- XACML:Condition, specifying further restrictions on the applicability of the rule beyond those specified in the *Target* and the *CredentialRequirements*;
- *DataHandlingPolicy*, describing how the information that needs to be revealed to satisfy this rule will be treated afterwards; this element is not defined in the original XACML language;
- *DataHandlingPreferences*, describing how the information contained in the resource that is protected by this rule has to be treated; this element is not defined in the original XACML language.

### 20.3.2 *Credential Requirements*

The policy language that we present is geared towards enabling technology-independent, user-centric and privacy-friendly access control on the basis of certified credentials. By a credential we mean an authenticated statement about attribute values made by an issuer, where the statement is independent from a concrete mechanism for ensuring authenticity. The statement made by the issuer is meant to affirm qualification. As credentials are not directly supported in the traditional policy languages,

---

<sup>1</sup> Rule combining algorithms provides the final Authorisation decision by combining the effects of all the rules in a policy, as: Deny-overrides: If one of the rules evaluates to Deny, then the final authorisation decision is Deny. Permit-overrides: If any rule evaluates to Permit, then the final authorisation decision is also Permit.

we extended the XACML Rule element such that credentials are the basic unit for reasoning about access control.

Each *Rule* can contain a *CredentialRequirements* element to specify the credentials that have to be presented in order to satisfy the *Rule*. The *CredentialRequirements* element contains a separate *Credential* element for each credential that needs to be presented. Each *Credential* element contains a unique identifier *CredentialId* that is used to refer to the credential from elsewhere in the *Rule*.

The *CredentialRequirements* element can also occur in parent *Policy* and *PolicySet* elements. They follow a typical distributive semantics; namely, one should treat the *CredentialRequirements* element of a *Rule* as if it contained all *Credential* elements specified within the rule itself, as well as those specified within all parent *Policy* and *PolicySet* elements.

### 20.3.3 Provisional Actions

The *ProvisionalActions* element is used to specify the provisional actions that a requester must perform before being granted access to the resource. Currently supported actions include revealing of attributes (to the Data Controller or to a Third Party) optionally under handling policy and credential proof, signing a statement, and so-called “spending” of credentials, which allows for placing restrictions on the number of times that the same credential is used to obtain access. Each action is described in a *ProvisionalAction* element; the language has to be extensible so that new types of *ProvisionalActions* can easily be added later on, and can refer to *DataHandlingPolicy* and *Credential* elements.

### 20.3.4 Data Handling Policies

The main purpose of the data handling policies is for the Data Controller to express what will happen to the information provided by the Data Subject during an access request. The provisional action to reveal an attribute value, for example, therefore contains an optional reference to the applicable *DataHandlingPolicy*. Each *Rule*, *Policy*, or *PolicySet* element can contain a number of *DataHandlingPolicy*. A *DataHandlingPolicy* can be referred to from anywhere in the rule by its unique *PolicyId* identifier.

A *DataHandlingPolicy* consists of a set of *Authorisations* that the Data Controller wants to obtain on the collected information, and a set of *Obligations*, that she promises to adhere to.

Before the Data Subject reveals her information, these *Authorisations* and *Obligation* are matched against the Data Subject’s *DataHandlingPreference* to see whether a matching *StickyPolicy* can be agreed upon.

### 20.3.5 Data Handling Preferences

The data *DataHandlingPreference* element defines how the information obtained from the resource protected by the Data Subject is to be treated after access is granted. The preferences are expressed by means of a set of *Authorisations* and *Obligations*, just like *DataHandlingPolicy* element. When access to the resource is requested, the *DataHandlingPreference* element has to be matched against a proposed *DataHandlingPolicy* to derive the applicable *StickyPolicy* - if a match can be found.

An important difference between *DataHandlingPreference* and *DataHandlingPolicy* is the resource that they pertain to. The *DataHandlingPreference* always describes how the resource protected by the Data Subject itself has to be treated after being collected. Whereas *DataHandlingPolicy* is used to communicate that a requester will have to reveal information in order to be granted access to the resource.

The main use of *DataHandlingPreference* is for a Data Subject to specify how she wants her PII to be treated by a Data Controller, i.e., which *Authorisations* she grants to the Data Controller with respect to her personal data, and which *Obligations* the Data Controller will have to adhere to.

Optionally, if the *DataHandlingPreference* contains a *AuthorisationDownstreamUsage*, this can be interpreted by optionally including a Policy specifying the downstream access control policy, i.e., the access control policy that has to be enforced on the downstream Data Controllers.

### 20.3.6 Sticky Policies

The *StickyPolicy* element is associated to a resource, meaning the agreed-upon sets of granted *Authorisations* and promised *Obligations* with respect to a resource. The *StickyPolicy* is usually the result of an automated matching procedure between the Data Subject's *DataHandlingPreference* and the Data Controller's *DataHandlingPolicy*.

The main difference between the *StickyPolicy* and the *DataHandlingPreferences* is that the former contains the *Authorisations* and *Obligations* that the policy-hosting entity itself has to adhere to, while the latter contains *Authorisations* and *Obligations* that an eventual recipient has to adhere to. Typically, a Data Subject will not impose on his or her self any *Authorisation* or *Obligation* concerning her own PII, so her policy will not contain a *StickyPolicy* element. The Data Controller, on the other hand, will describe in the *StickyPolicy* the *Authorisation* and *Obligation* that she, herself, has to adhere to, while the *DataHandlingPreferences* contain those that a Downstream Data Controller has to adhere to. Usually, the Downstream Data Controller (Third Party) will be subject to the same or stronger restrictions than the Data Controller herself, meaning that the policy specified in the *DataHandlingPreference*

*ences* will usually be at most as permissive as the policy specified in the *StickyPolicy*.

### 20.3.7 Obligations

The elements *DataHandlingPolicy*, *DataHandlingPreferences*, and *StickyPolicy* contain a set of *Obligation* elements. An *Obligation* is defined as: “A promise made by a Data Controller to a Data Subject in relation to the handling of her PII. The Data Controller is expected to fulfill the promise by executing and/or preventing a specific action after a particular event, e.g., time, and optionally under certain conditions.”

As defined previously, an obligation is often defined as Event-Condition-Action:

*On Event If Condition Do Action.*

For facilitating the comparison of obligations, we consider *Triggers* as events filtered by conditions. In other words, we replace the notions of events and conditions by *Trigger*. The *Triggers* are events that are considered by an obligation and can be seen as the set of events that result in actions.

Obligations in PPL language, consists of a set of *Obligation* elements. This latter defines a set of *Triggers* describing the events that trigger the obligation, and the related *Action* that has to be performed.

The reason why we did not choose to use the standard *XACML:Obligations* element to specify the obligations (in which we embed the data handling policies or preferences), is that XACML Obligations can only be used to specify obligations that the PEP has to adhere to when an access request occurs. This cannot be used for our data handling policies, since the latter pertain to information that the requester will have to reveal in order to obtain access, rather than to the resource being protected. This XACML element cannot be reused for our data handling preferences either, since the latter specifies obligations that the recipient of the resource has to adhere to, rather than the PEP that is protecting access to the resource. In other words, by populating the *XACML:Obligations* element that protects her personal data, a Data Subject would impose obligations that she herself has to adhere to each time a Data Controller requests access to the personal data, rather than imposing obligations on the Data Controller.

Since obligations triggered by access requests are only a small subclass of the obligations that we consider here, we chose to leave the storage and enforcement of obligations entirely up to the obligation Engine, and let the PEP simply signal the obligation Engine each time an access request occurs.

### 20.3.8 Authorisations

*DataHandlingPolicy*, *DataHandlingPreference*, and *StickyPolicies* contain, apart from the set of *Obligations* described above, also a set of *Authorisations*. While obligations specify actions that the Data Controller is required to perform on the transmitted information, authorisations specify actions that it is allowed to perform. Similarly to what we did for obligations, we recognise that it is impossible to define an exhaustive list of authorisations that covers all needs that may ever arise in the real world. Rather, we define a generic, user-extensible structure for authorisations so that new, possibly industry-specific authorisation vocabularies can be added later on. We do provide, however, a basic authorisation vocabulary for using data for certain purposes and for downstream access control (to forward the information to third parties), and we describe how these authorisations can be efficiently matched via a general strategy.

- **Authorisation Purposes:** The first concrete authorisation type that we define is the authorisation to use information for a particular set of purposes. The *AuthUseForPurpose* elements are referred to by standard URIs specified in agreed-upon vocabularies of usage purposes. These vocabularies of URIs may be organised as flat lists or as hierarchical ontologies.
- **Authorisation for downstream usage:** Called *AuthDownStreamUsage*, is the second concrete authorisation type that we define as the authorisation to forward the information to third parties, so-called downstream Data Controllers. Optionally, this authorisation enables the Data Subject to specify the access control policy under which the information will be made available, i.e., the minimal access control policy that the (primary) data controller has to enforce when sharing the information with downstream Data Controllers.

## 20.4 PPL Engine Data Model

In this section, we define all the implementation details of the PPL engine. We are using class and package diagrams to describe in detail how the different components of the engine are implemented. In order to facilitate the manipulation of the different elements of the XML policy, we decided to map all these elements into Java classes. These classes are then stored into the persistence database and called as soon as we need to read, modify or generate a new policy. For example, if we want to generate a sticky policy, after matching a privacy policy and a preference, we call all the classes related to the elements of this sticky policy and we generate an XML file. This method is less complex than the selection and assembling of pure XML elements. In this chapter, all the PPL language elements are describes as Java classes.

In the following sections, we define the PPL prefix of one element to define that it as a PrimeLife element, and the XACML prefix for the XACML elements.



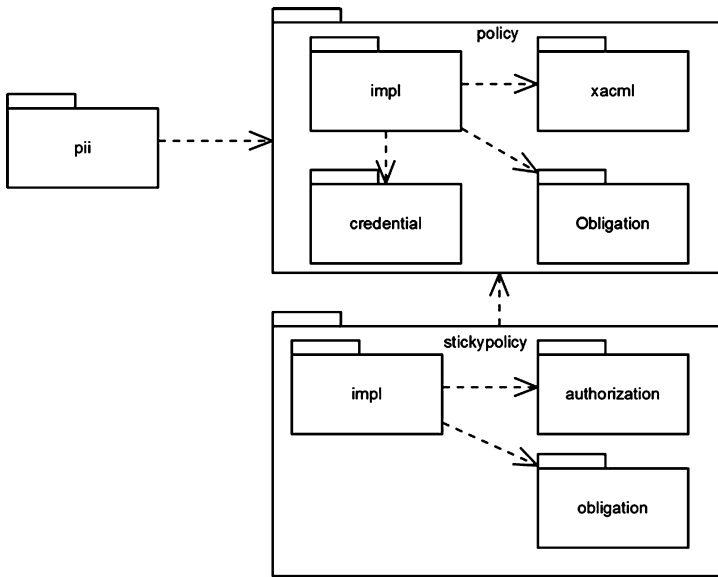


Fig. 20.4: Data model package diagram representing the dependencies between the packages.

### 20.4.1 Package *pii*

The package *pii* contains the data class to represent the PII. It is constituted by the *PIIType* class. The *PPIType* is used to represent the Personally Identifiable Information (PII) in a simple way.

This class is composed of:

- The *AttributeName* element, describing the name identifier of the PII, for example, <http://www.w3.org/2006/vcard/ns/#email> which indicates the email PII, or also <http://www.fgov.be/eID/address> to indicates the address information;
- The *AttributeValue* element, describing the value of the PII, for example if we consider the previous *AttributeName* examples, we can have [mail@mail.com](mailto:mail@mail.com) as a value corresponding to the <http://www.w3.org/2006/vcard/ns/#email> *AttributeValue*;
- *CreationDate* and *DateModification*, describing extra date information to the user.

### 20.4.2 Package *policy.Impl*

This package contains all the data classes to represent the skeleton of the language data structure.

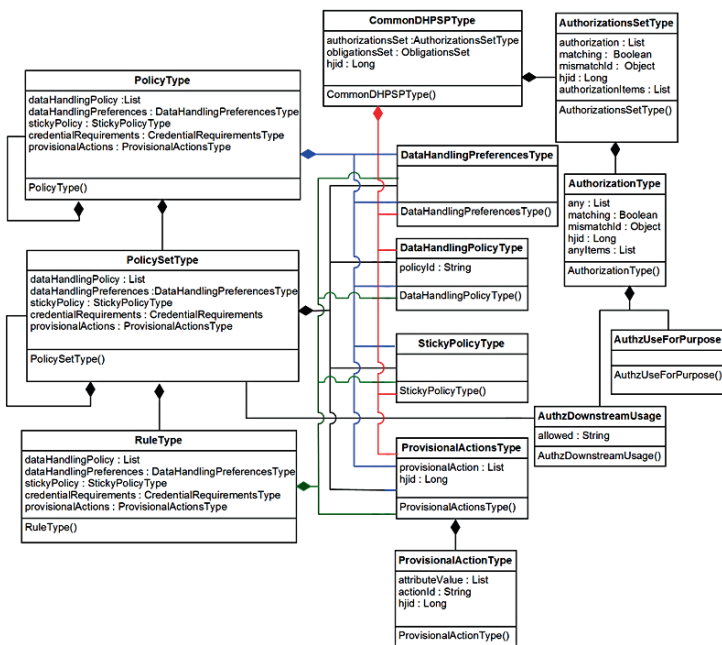


Fig. 20.5: Simplified policy data model class diagram.

At the high level of the tree structure, we have the PPL *PolicySetType* and PPL *PolicyType* elements that successively extend the XACML *PolicySet* and the XACML *Policy* classes. The latter are present in the XACML package, and the former implements the *PPLEvaluatable* interface. We used the *PPLEvaluatable* interface to provide a generic way to define a *Policy*, because a *Policy* can be defined by either the *PolicySetType* class or by the *PolicyType* class.

The *PolicySetType* class is a top level element. It is an aggregation of other *PolicySetTypes* and *PolicyTypes*. And the *PolicyType* class is an aggregation of other *PolicyTypes* and *RuleTypes* elements.

The *PolicySetType*, *PolicyType* and *RuleType* classes are composed of a different class; *CommonDHPSPTType*, which is a generic class for the *DataHandlingPolicy*, *DataHandlingPreferences* and *StickyPolicy* classes, *CredentialRequirements* class and *ProvisionalActions* class.

As the *DataHandlingPolicyType*, the *DataHandlingPreferencesType* and the *StickyPolicyType* classes represent the same data structure, and they are only different from the interpretation meaning, they extend the generic class, *CommonDHPSPType*.

The *CommonDHPSPType* consists of a set of authorisations, defined by the *AuthorisationsSetType* class, and a set of obligations, expressed in the *ObligationsSet* class (defined in the next section).

The *AuthorisationsSetType* class is composed of a set of authorisations defined by the abstract class *AuthorisationType*. This is due to the fact that our language supports an extensible authorisation vocabulary, but we predefine two concrete authorisation types here. The first is the authorisation to use the information for a list of purposes, enumerated inside the *AuthzUseForPurpose* class. The second predefined authorisation type, *AuthzDownstreamUsage*, contains a *Boolean* attribute indicating whether downstream usage is allowed or not in association with a *PolicyType* attribute that represents the policy preferences for the third party.

The *ProvisionalActionsType* class is composed of a set of *ProvisionalActionType*. This latter describes a single provisional action that needs to be fulfilled in order to satisfy a rule. The *ProvisionalActionType* class contains the *ActionId* attribute and a set of XACML *AttributeValue*. The *ActionId* attribute represents the identifier (URI) of the action to be performed. The set of the XACML *AttributeValue* represents arguments of the action, which may include other functions. The semantics of the argument depend on the particular action being performed. Some actions are defined:

- <http://www.primelife.eu/Reveal>: This action requires the Data Subject to reveal an attribute. The attribute could be part of one of her credentials, or could be a self stated, uncertified attribute. The action takes one or two arguments of data-type <http://www.w3.org/2001/XMLSchema#anyURI>. The first (mandatory) argument is the URI of the attribute to be revealed. The second (optional) argument is a URI referring to the credential identifier (*CredentialID*) of the credential that contains the attribute.
- <http://www.primelife.eu/RevealUnderDHP>: This action requires the Data Subject to reveal an attribute while specifying the data handling policy that will be applied to the attribute after it is revealed. The attribute could be part of one of her credentials, or could be a self-stated, uncertified attribute. The action takes two or three arguments of data-type <http://www.w3.org/2001/XMLSchema#anyURI>. The first (mandatory) argument is the URI of the attribute to be revealed. The second (mandatory) argument is a URI referring to the data handling policy under which the attribute has to be revealed. The third (optional) argument is a URI referring to the credential identifier of the credential that contains the attribute
- <http://www.primelife.eu/RevealTo>: This action requires the requester to reveal an attribute to an external third party. The attribute could be part of

one of her credentials, or could be a self-stated, uncertified attribute. The action takes two or three arguments of data-type <http://www.w3.org/2001/XMLSchema#anyURI>. The first (mandatory) argument is the URI of the attribute to be revealed. The second (mandatory) argument is the URI defining the third party to whom the attribute will be revealed. The third (optional) argument is a URI referring to the credential identifier of the credential that contains the attribute.

- <http://www.primelife.eu/RevealToUnderDHP>: This action requires the Data Subject to reveal an attribute to an external third party while specifying the data handling policy that will be applied to the attribute after it is revealed. The attribute could be part of one of her credentials, or could be a self-stated, uncertified attribute. The action takes three or four arguments of data-type <http://www.w3.org/2001/XMLSchema#anyURI>. The first (mandatory) argument is the URI of the attribute to be revealed. The second (mandatory) argument is the URI defining the third party to whom the attribute will be revealed. The third (mandatory) argument is a URI referring to the data handling policy under which the attribute has to be revealed. The fourth (optional) argument is a URI referring to the credential identifier of the credential that contains the attribute.
- <http://www.primelife.eu/Sign>: This action requires the requester to sign a statement before accessing the resource. How the signature is implemented depends on the underlying technology, but carries the semantics that a verifier can check later that a particular Data Subject satisfying the policy explicitly agreed to the statement. The action takes a single argument of data-type <http://www.w3.org/2001/XMLSchema#string> describing the statement that needs to be signed.
- <http://www.primelife.eu/Spend>: This action requires the requester to “spend” one of her credentials, thereby imposing restrictions on how many times the same credential can be used in an access request. The action takes four mandatory arguments. The first is of data-type <http://www.w3.org/2001/XMLSchema#anyURI> and contains the CredentialId of the credential that has to be spent. The second and third arguments are of type <http://www.w3.org/2001/XMLSchema#integer>. The second argument is the number of units that have to be spent for this access; the latter is the spending limit, i.e., the maximum number of units that can be spent with this credential on the same scope. The fourth argument is of data-type <http://www.w3.org/2001/XMLSchema#string> and defines the scope on which the credential has to be spent.

### 20.4.3 Package Credential

The *CredentialRequirementsType* class is composed of a set of credentials and conditions. Each individual credential is a condition within a credential.

The PPL *ConditionType* class contains an abstract attribute of type XACML *Expression*. This latter class provides a way to define an obligation within a credential.

The *CredentialType* class is used to declare a credential that has to be held by an access requester. It contains a *CredentialId* attribute that identifies the credential element and is in relation of 1..0 with the *UndisclosedExpressionType* class. This class acts as a placeholder to indicate that a credential condition was omitted due to policy sanitization.

Also, the *CredentialType* class is related to the *AttributeMatchAnyOfType* class. This class can specify conditions directly within the *CredentialType* class.

The *AttributeMatchAnyOfType* class is used for matching a given attribute with a list of values, whereby for every list element an individual matching algorithm is used.

Although in principle any attribute can be matched, the *AttributeMatchAnyOfType* construction is particularly useful for providing lists of accepted credential types or issuers. Clearly, if no credential types are explicitly specified, then any credential type that contains the necessary attributes can be used to satisfy the policy. If no issuers are satisfied, then credentials by any issuer are accepted.

The element *AttributeMatchAnyOfType* class contains the following attributes; *AttributeId*, which determines the name of the attribute in this credential that is matched against the list of values, *disclose attribute*, the type of policy disclosure used for this element when this policy is sent to the Data Subject. Possible values are “yes,” “no,” and “attributes-only.” When the attribute is omitted, the default value “yes” is assumed.

When set to “yes,” this *AttributeMatchAnyOf* element is sent unmodified to the Data Subject. When set to “no,” this *AttributeMatchAnyOf* element is sanitised by means of the following substitutions:

- The value of *AttributeId* is replaced with “undisclosed”
- Each *MatchValue* child element within this *AttributeMatchAnyOf* element is replaced with an *UndisclosedExpression* element.

When set to “attributes-only,” then only the latter substitution is performed, i.e., all *MatchValue* child elements are replaced with an *UndisclosedExpression* element. (See the section policy sanitisation).

The *MatchValueType* class defines a literal value against which the given attribute (specified with *AttributeId* in the parent *AttributeMatchAnyOf* element) is matched, as well as the matching algorithm that is used. The class contains the following attributes:

- *MatchId* that indicates the name of the matching algorithm that is used to match the attribute with the literal.

- *DataType* attribute of the literal value against which the attribute will be matched
- *Disclose* attribute that defines the type of policy disclosure used for this element when this policy is sent to the Data Subject. Possible values are “yes” and “no.”

20.4.4 Package Obligations

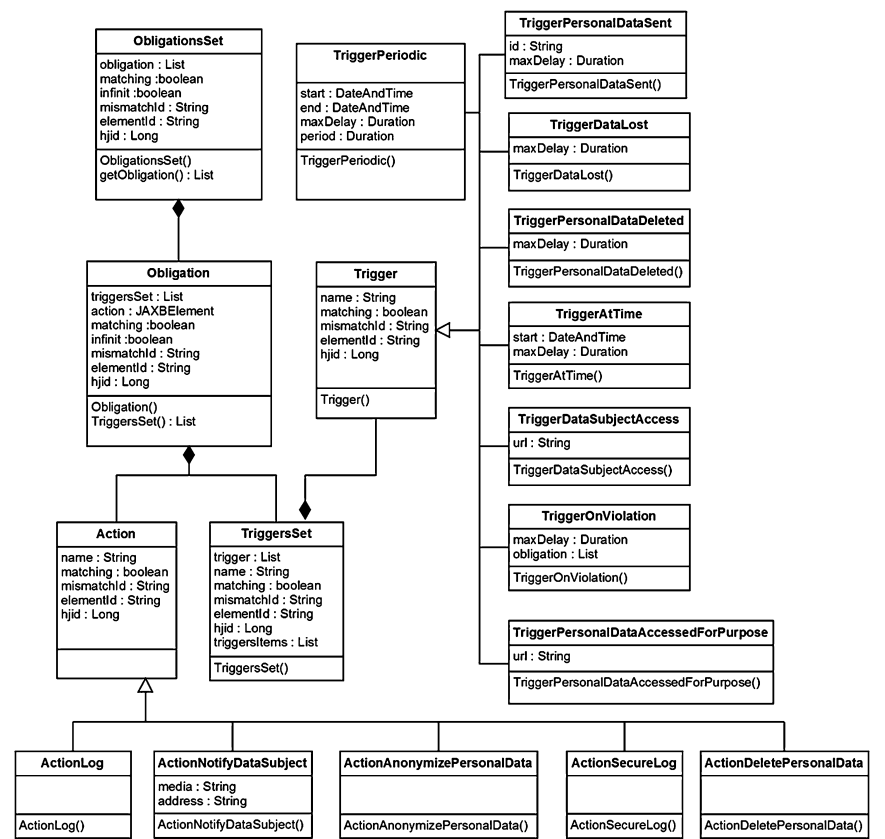


Fig. 20.6: Simplified obligation data model class diagram.

The obligation package contains all of the data classes to define obligations. The main class in this package is *ObligationsSet* class, which is composed of a set of *Obligation* objects. Each *Obligation* contains one *TriggerSet*, which in turn contains

a set of *Trigger* objects describing the events that trigger the obligation, and one *Action* element defining the action that has to be performed.

There are different types of triggers that extend the abstract class *Trigger*, and different types of actions that extend the abstract class *Action*. The language and the design for defining obligation may be slightly different to express obligations in the Data Controller's privacy policy, in the Data Subject's privacy preferences, and in sticky policies:

- The Data Subject's privacy preferences specify "required obligations," i.e. what the Data Subject requires in terms of obligations to provide a given piece of personal data to a given Data Controller.
- The Data Controller's privacy policy specifies "proposed obligations," i.e. what the Data Controller is willing (and able) to enforce in terms of obligations for a given collected data.
- The sticky policy specifies "committed obligations," i.e. the obligations the Data Subject and the Data Controller agreed upon and that must be enforced by the Data Controller.

Here is a brief description of some of the common triggers and actions:

- **Trigger at Time:** A time-based trigger that occurs only once between start and start + maxDelay.
- **Trigger Periodic:** A time-based trigger that occurs multiple times on a periodic basis between start and end.
- **Trigger Personal Data Accessed for Purpose:** An event-based trigger that occurs each time the personal data associated with the obligation is accessed for one of the specified purposes.
- **Trigger Personal Data Sent:** An event-based trigger that occurs when the PII associated with the obligation is shared with a third party (downstream Data Controller).
- **Action *DeletePersonalData*:** This action deletes a specific piece of information, and is intended for handling data retention.
- **Action *NotifyDataSubject*:** This action notifies the Data Subject when triggered, i.e. send the trigger information to the Data Subject.
- **Action *Log*:** This action logs an event, e.g., write in a trace file the trigger information.

#### **20.4.5 Package *StickyPolicy***

The *StickyPolicy* package contains all of the data classes to define the result of the matching process. This package is composed of three sub-packages.

### 20.4.5.1 Sub-package impl

The sub-package impl of the sticky policy package contains the data structure skeleton of the matching process result. The *StickyPolicy* class represents the main component. It contains matching attributes, which indicate if the final matching is true (there is a match) or not (there is a mismatch), and is composed by different *AttributeType* elements.

The *AttributeType* class represents the data type of a matching PII. So, if in the policy we have three revealing actions, under a particular data handling policy, of three PII, we will have three *AttributeType* objects within the *StickyPolicy* object. This class contains, as for the *StickyPolicy* class:

- A *Matching* attribute to indicate if the PII has a matching or not.
- An *attributeURI* element to represent the attribute name value of the PII, and composed of:
  - An *authorisationsSet* element, which represents the *authorisationSet StickyPolicy* of the matching;
  - An *obligationsSet* element, which represents the *obligationsSet StickyPolicy* of the matching process, and a mismatched element in case there is a mismatch as a result of the matching.
- The *MismatchesType* class contains two types; *authorisationsMismatch* and *obligationsMismatch*. These elements are only present if a mismatch of the corresponding type occurs. Each *MismatchType* is defined in a separate package, because the PPL language is extensible, and the definition of the mismatch depends of the *DataHandling* type.

### 20.4.5.2 Sub-package Authorisations Mismatch

The sub-package *Authorisation* represents the data classes of the authorisations mismatch. The *AuthorisationsMismatchType* class represents the main class. It contains a *mismatchId* attribute that is used to be referred within the *AuthorisationsSet* element, and composed of either the *AuthorisationsSetMismatchType* or with the two *AuthzUseForPurposeType* and *AuthzDownStreamUsageType* together (or only one of the elements).

The *Mismatch* class is defined to help the engine and the UI to display the mismatching elements and permit to the user to make a decision without being obliged to come back to his preferences and compare it with the sticky policy.

We distinguish the authorisation and the obligation mismatches using the two classes *AuthzUseForPurposeType* and *AuthzDownStreamUsageType*. In some cases, we can have either an authorisation use for purpose mismatch, or an authorisation downstream usage mismatch, or both together. To notify and describe the mismatch, we use the same concept mentioned above, we indicate the policy values (which represent the Data Controller values) and the preferences values (which represent the Data Subject values).



### **20.4.5.3 Sub-package Obligations Mismatch**

To notify and describe the mismatch, we use the same concept mentioned previously in the authorisation mismatch. We use the matching attribute to indicate whether a match occurs or not, and we indicate the policy values (which represent the Data Controller values) and the preferences values (which represent the Data Subject values), in case of matching.

## **20.5 Conclusion**

In this chapter, we document and describe how we implemented the PPL engine in charge of interpreting and enforcing policies and preferences defined by the Data Controllers and the Data Subjects. We detailed the symmetric architecture supporting the PPL engine by explaining how each component is working and interacting with the other one. This chapter describes mainly the implementation options chosen to develop the engine according to the requirement defined in the previous chapters.

## References Part IV

- [ABD<sup>+</sup>10] Claudio A. Ardagna, Laurent Bussard, Sabrina De Capitani Di, Gregory Neven, Stefano Paraboschi, Eros Pedrini, Stefan Preiss, Dave Raggett, Pierangela Samarati, Slim Trabelsi, and Mario Verdicchio. Primelife policy language. 2010.
- [ACDS08] C.A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. A privacy-aware access control system. *Journal of Computer Security (JCS)*, 16(4):369–397, 2008.
- [ACK<sup>+</sup>10] Claudio Agostino Ardagna, Jan Camenisch, Markulf Kohlweiss, Ronald Leenes, Gregory Neven, Bart Priem, Pierangela Samarati, Dieter Sommer, and Mario Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the PRIME Project. *Journal of Computer Security*, 18(1):123–160, 2010.
- [ADF<sup>+</sup>10a] C.A. Ardagna, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, and P. Samarati. Minimizing disclosure of private information in credential-based interactions: A graph-based approach. In *Proc. of the 2nd IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT 2010)*, Minneapolis, MN, USA, August 2010.
- [ADF<sup>+</sup>10b] C.A. Ardagna, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, and P. Samarati. Supporting privacy preferences in credential-based interactions. In *Proc. of the ACM Workshop on Privacy in the Electronic Society (WPES 2010)*, Chicago, IL, USA, October 2010.
- [ADN<sup>+</sup>10] Claudio A. Ardagna, Sabrina De Capitani di Vimercati, Gregory Neven, Stefano Paraboschi, Franz-Stefan Preiss, Pierangela Samarati, and Mario Verdicchio. Enabling privacy-preserving credential-based access control with XACML and SAML. In *10th IEEE International Conference on Computer and Information Technology (CIT 2010)*, pages 1090–1095. IEEE Computer Society, 2010.
- [ADP<sup>+</sup>10] Claudio A. Ardagna, Sabrina De Capitani di Vimercati, Stefano Paraboschi, Eros Pedrini, Pierangela Samarati, and Mario Verdicchio. Expressive and deployable access control in open web service applications. *IEEE Transaction on Services Computing*, 2010, to appear.
- [AHKS02] P. Ashley, S. Hada, G. Karjoth, and M. Schunter. E-P3P privacy policies and privacy authorization. In *Proc. of the ACM Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.
- [APE05] APEC. Chapter ii and viii of the apec privacy framework. [http://www.apec.org/apec/newsmedia/factsheets/apecprivacyframe-work.MedialibDownload.v1\(21.109\)](http://www.apec.org/apec/newsmedia/factsheets/apecprivacyframe-work.MedialibDownload.v1(21.109)), 2005.
- [BCC05] E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *Proc. of the 11th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
- [BCS05] M. Backes, J. Camenisch, and D. Sommer. Anonymous yet accountable access control. In *Proc. of the ACM Workshop on Privacy in the Electronic Society (WPES 2005)*, Alexandria, VA, USA, November 2005.

- [BDDS01] P. Bonatti, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. An access control model for data archives. In *Proc. of the 16th International Conference on Information Security*, Paris, France, June 2001.
- [BFIK98] M. Blaze, J. Feigenbaum, J. Ioannidis, and A.D. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, 1998.
- [BFL96] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 1996.
- [BMB09] Moritz Y. Becker, Alexander Malkis, and Laurent Bussard. MSR-TR-2009-128: A framework for privacy preferences and data-handling policies. Technical report, Microsoft Research, September 2009.
- [Bro08] K. Brown. "the infocard identity revolution". [http://technet.microsoft.com/enus/magazine/cc160966\(printer\).aspx](http://technet.microsoft.com/enus/magazine/cc160966(printer).aspx), 2008.
- [BS02] P. Bonatti and P. Samarati. A unified framework for regulating access and information release on the web. *Journal of Computer Security (JCS)*, 10(3):241–272, 2002.
- [Bun08] Deutscher Bundestag. *Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), das zuletzt durch Artikel 1 des Gesetzes vom 14. August 2009 (BGBl. I S. 2814) geändert worden ist*. Bundesanzeiger Verlag, 2008.
- [CCKT05] W. Chen, L. Clarke, J. Kurose, and D. Towsley. Optimizing cost-sensitive trust-negotiation protocols. In *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Miami, FL, USA, March 2005.
- [CD00] J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Proc. of the 6th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2000)*, Kyoto, Japan, September 2000.
- [CDFS07a] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.
- [CDFS07b] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Microdata protection. In T. Yu and S. Jajodia, editors, *Secure Data Management in Decentralized Systems*. Springer-Verlag, 2007.
- [CDFS08] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-anonymous data mining: A survey. In C.C. Aggarwal and P.S. Yu, editors, *Privacy-Preserving Data Mining: Models and Algorithms*. Springer-Verlag, 2008.
- [CFL<sup>+</sup>97] Y-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. Referee: Trust management for web applications. *Computer Networks and ISDN Systems*, 29(8–13):953–964, 1997.
- [CGP<sup>+</sup>08] S. Cimato, M. Gamassi, V. Piuri, R. Sassi, and F. Scotti. Privacy-aware biometrics: Design and implementation of a multimodal verification system. In *Proc. of the Annual Computer Security Applications Conference (ACSAC 2008)*, Anaheim, CA, USA, December 2008.
- [Cha85] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, October 1985.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.
- [CL08] David Chadwick and Stijn Lievens. Enforcing “sticky” security policies throughout a distributed application application. In *1st International Workshop on Middleware Security (Midsec2008)*, 2008.
- [CMN<sup>+</sup>10] Jan Camenisch, Sebastian Mödersheim, Gregory Neven, Franz-Stefan Preiss, and Dieter Sommer. A card requirements language enabling privacy-preserving access control. In James B. D. Joshi and Barbara Carminati, editors, *15th ACM Symposium on Access Control Models and Technologies (SACMAT 2010)*, pages 119–128. ACM, 2010.

- [Com95] European Commission. Art. 7 of the data protection directive 95/46/ec, 1995.
- [Con02] ContentGuard. XrML 2.0 Technical Overview. <http://www.xrml.org/reference/XrMLTechnicalOverviewV1.pdf>, 2002.
- [Con07a] PRIME Consortium. HCI guidelines. <https://www.primeproject.eu/primeproducts/reports/arch/pubdelD06.1.fecwp06.1v1final.pdf>, 2007.
- [Con07b] PRIME Consortium. PRIME data model. <https://www.primeproject.eu/ont/Data-model.html>, 2007.
- [Cou83] German Supreme Court. Bverfge , 65,1 (volkszählung, az.1 bvr 209). *Entscheidungen des Bundesverfassungsgerichts*, 65:1, 1983.
- [Cra02] L.F. Cranor. *Web Privacy with P3P*. O'Reilly & Associates, 2002.
- [CSBS] Ryan Calo, Max Senges, Andreas Braendhagen, and Jan Schallaböck. Privicons - privacy icons for email usage. <http://privicons.org/>.
- [CSF<sup>+</sup>08] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard), May 2008. <http://www.ietf.org/rfc/rfc5280.txt>.
- [CV02] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proc. of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, USA, November 2002.
- [DFJS07] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, and P. Samarati. Access control policies and languages. *International Journal of Computational Science and Engineering (IJCSSE)*, 3(2), 2007.
- [Dir95] Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal L 281*, pages 31–50, 23/11/1995.
- [eXt05] *eXtensible Access Control Markup Language (XACML) Version 2.0*, February 2005. [http://docs.oasis-open.org/xacml/2.0/access/\\_control-xacml-2.0-core-spec-os.pdf](http://docs.oasis-open.org/xacml/2.0/access/_control-xacml-2.0-core-spec-os.pdf).
- [Fac10] Profile options, 2010.
- [Fis06] J. Fishenden. "creative commons and its wider potential". 2006.
- [FK92] D. Ferraiolo and R. Kuhn. Role-based access control. In *Proc. of the 15th NIST-NCSC National Computer Security Conference*, pages 554–563, October 1992.
- [GD06] S. Gevers and B. De Decker. Automating privacy friendly information disclosure. *Technical Report CW441, K.U. Leuven, Dept. of Computer Science*, April 2006.
- [GLM<sup>+</sup>04] M. Gamassi, M. Lazzaroni, M. Misino, V. Piuri, D. Sana, and F. Scotti. Accuracy and performance of biometric systems. In *Proc. of the 21st IEEE Instrumentation and Measurement Technology Conference (IMTC 2004)*, Como, Italy, May 2004.
- [GPSS05] M. Gamassi, V. Piuri, D. Sana, and F. Scotti. Robust fingerprint detection for access control. In *Proc. of RoboCare Workshop 2005*, Rome, Italy, May 2005.
- [HBP05] M. Hilty, D. Basin, and A. Pretschner. On obligations. *Lecture Notes in Computer Science*, 3679:98–117, 2005.
- [HPB<sup>+</sup>07] Manuel Hilty, Alexander Pretschner, David Basin, Christian Schaefer, and Thomas Walter. A policy language for distributed usage control. In Joachim Biskup and Javier Lopez, editors, *12th European Symposium on Research in Computer Security (ESORICS 2007)*, volume 4734 of *LNCS*, pages 531–546. Springer-Verlag, 2007.
- [IDE] IDENTITY MIXer (IDEMIX). <http://www.zurich.ibm.com/security/idemix/>.
- [IY05] K. Irwin and T. Yu. Preventing attribute information leakage in automated trust negotiation. In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
- [KOB08] P. Kärger, D. Olmedilla, and W.-T. Balke. Exploiting preferences for minimal credential disclosure in policy-driven trust negotiations. In *Proc. of the 5th VLDB Workshop on Secure Data Management (SDM 2008)*, Auckland, New Zealand, August 2008.
- [Kun07] Christopher Kuner. *European Data Protection Law*, volume 2nd Edition. Oxford University Press, 2007.

- [LGF00] N. Li, B.N. Grosz, and J. Feigenbaum. A practically implementable and tractable delegation logic. In *Proc. of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, June 2000.
- [Lin10] LinkedIn. "user agreements". [http://www.linkedin.com/static?key=useragreementtrk=hbf\\_tuserag](http://www.linkedin.com/static?key=useragreementtrk=hbf_tuserag), 2010.
- [LMW05] N. Li, J.C. Mitchell, and W.H. Winsborough. Beyond proof-of-compliance: Security analysis in trust management. *Journal of the ACM*, 52(3):474–514, 2005.
- [Luh77] N. Luhmann. Differentiation of society. *Canadian Sociological Review*, 2:29–53, 1977.
- [LWBW08] A. J. Lee, M. Winslett, J. Basney, and V. Welch. The Traust authorization service. *ACM Transactions on Information and System Security (TISSEC)*, 11(1):1–33, February 2008.
- [Nis98] Helen F. Nissenbaum. Protecting privacy in an information age: The problem of privacy in public. *Law and Philosophy Vol 17*, pp. 559-596, 1998.
- [Nis04] H. F. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, Vol. 79, No. 1, 2004.
- [Nis10] H. F. Nissenbaum. Privacy in context: Technology, policy and the integrity of social life. *Stanford Law Books* 65, 2010.
- [NLW05] J. Ni, N. Li, and W.H. Winsborough. Automated trust negotiation using cryptographic credentials. In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
- [OAS05a] OASIS. Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0, 2005. Available from: <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.
- [OAS05b] OASIS. SAML 2.0 profile of XACML v2.0. OASIS Standard, 2005.
- [OAS08] OASIS. Oasis extensible access control markup language (xacml) tc. <http://www.oasis-open.org/committees/tchome.php?wgabbrev=xacml>, 2008.
- [ODR02] ODRL. Open Digital Rights Language (ODRL), version 1.1, 2002.
- [OEC80] OECD. *Oecd guidelines on the protection of privacy and transborder flows of personal data*. OECD, 1980.
- [Ope07] OpenID authentication 2.0, December 2007. <http://openid.net/developers/specs/>.
- [Par04] Article 29 Working Party. *WP 100 Opinion on more harmonised information provisions*. European Commission, 2004.
- [Pri09a] PrimeLife WP5.1. First research report on research on next generation policies. In Pierangela Samarati, editor, *PrimeLife Deliverable D5.2.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2009.
- [Pri09b] PrimeLife WP5.2. Draft 2nd design for policy languages and protocols. In Dave Raggett, editor, *PrimeLife Heartbeat H5.3.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, July 2009.
- [Pri09c] Dynamic Coalition Privacy. Privacy-rights-agreements, 2009.
- [Pri11a] PrimeLife WP5.2. Report on research on legal policy mechanisms. In Leif-Erik Holtz and Jan Schallabek, editors, *PrimeLife Heartbeat H5.2.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2011.
- [Pri11b] PrimeLife WP5.2. Third research report on research on next generation policies. In Sabrina De Capitani di Vimercati and Pierangela Samarati, editors, *PrimeLife Deliverable D5.2.3*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2011.
- [Pro07] Liberty Alliance Project. Identity governance, 2007.
- [PS04] Jaehong Park and Ravi Sandhu. The UCONABC usage control model. *ACM Trans. Inf. Syst. Secur.*, 7(1):128–174, 2004.
- [PSSW08] A. Pretschner, F. Schütz, C. Schaefer, and T. Walter. Policy evolution in distributed usage control. In *4th Intl. Workshop on Security and Trust Management*. Elsevier, June 2008.

- [Run06] M. Rundle. International data protection and digital identity management tools (using icons to express user preferences). Presentation at IGF2006 PrivacyWorkshop 1, Athens 2006, 2006.
- [RZN<sup>+</sup>05] T. Ryutov, L. Zhou, C. Neuman, T. Leithead, and K.E. Seamons. Adaptive trust negotiation and access control. In *Proc. of the 10th ACM Symposium on Access Control Models and Technologies*, Stockholm, Sweden, June 2005.
- [SCFY96] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [SD01] P. Samarati and S. De Capitani di Vimercati. Access control: Policies, models, and mechanisms. In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, volume 2171 of *LNCS*. Springer-Verlag, 2001.
- [SWW97] K. E. Seamons, W. Winsborough, and M. Winslett. Internet credential acceptance policies. In *Proc. of the Workshop on Logic Programming for Internet Applications*, Leuven, Belgium, July 1997.
- [SWY01] K. Seamons, M. Winslett, and T. Yu. Limiting the disclosure of access control policies during automated trust negotiation. In *Proc. of the Network and Distributed System Security Symposium (NDSS 2001)*, San Diego, CA, USA, April 2001.
- [U-P07] Credentica. *U-Prove SDK overview: A Credentica white paper*, 2007. <http://www.credentica.com/files/U-ProveSDKWhitepaper.pdf>.
- [W3C01] P3p v1.1. 2001.
- [W3C02] W3C. A P3P preference exchange language 1.0 (APPEL1.0), 2002.
- [W3C03] W3C. Enterprise privacy authorization language (epal 1.2). <http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/>, 2003.
- [W3C06a] W3C. The platform for privacy preferences 1.1 (P3P1.1) specification, 2006.
- [W3C06b] W3C. Web services policy 1.2 - framework (ws-policy). <http://www.w3.org/Submission/WS-Policy/>, 2006.
- [Wan04] Xin Wang. Mpeg-21 rights expression language: Enabling interoperable digital rights management. *IEEE MultiMedia*, 11(4):84–87, 2004.
- [WCJS97] M. Winslett, N. Ching, V. Jones, and I. Slepchin. Assuring security and privacy for digital library transactions on the web: Client and server security policies. In *Proc. of the 4th International Forum on Research and Technology Advances in Digital Libraries (ADL '97)*, Washington, DC, USA, May 1997.
- [Web06] Web services policy framework. [http://www.ibm.com/developerworks/webservices/library/specification/ws-polfram/?S\\\_TACT=105AGX04\\$&\\$S\\\_CMP=LP](http://www.ibm.com/developerworks/webservices/library/specification/ws-polfram/?S\_TACT=105AGX04$&$S\_CMP=LP), March 2006.
- [WSJ00] W. Winsborough, K. E. Seamons, and V. Jones. Automated trust negotiation. In *Proc. of the DARPA Information Survivability Conference & Exposition (DISCEX 2000)*, Hilton Head Island, SC, USA, January 2000.
- [WWJ04] L. Wang, D. Wijesekera, and S. Jajodia. A logic-based framework for attribute based access control. In *Proc. of the ACM Workshop on Formal Methods in Security Engineering (FMSE 2004)*, Washington, DC, USA, October 2004.
- [YFAT08] D. Yao, K.B. Frikken, M.J. Atallah, and R. Tamassia. Private information: To reveal or not to reveal. *ACM Transactions on Information and System Security (TISSEC)*, 12(1):1–27, October 2008.
- [YW03] T. Yu and M. Winslett. A unified scheme for resource protection in automated trust negotiation. In *Proc. of the IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 2003.
- [YWS03] T. Yu, M. Winslett, and K.E. Seamons. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust. *ACM Transactions on Information and System Security (TISSEC)*, 6(1):1–42, February 2003.



**Part V**  
**Infrastructures for Privacy and Identity**  
**Management**



## Introduction

The establishment of identity management infrastructures on a global scale is notoriously difficult. Microsoft Passport has failed here, in spite of a hard-to-match installed client, allegedly for reasons of trust and privacy [KR00]. Identity management platforms today hardly span domains and only bring together a handful of services. Lack of trust in any given organisation, technological problems with security and privacy and compliance and liability issues are only some of the obstacles to the establishment of a more global identity management system [BHTB05].

In the course of, e.g., the PRIME project [CLS11], it became clear that:

- Business models for privacy and privacy-enhancing IdM are not trivial.
- Infrastructure aspects are often overlooked, while having a significant impact on the adoption of IdM solutions, security and privacy functionality of IdM systems in general, and specifically privacy enhancing Identity Management Systems.

Also, a wide range of protocols and implementations are available in this field, making the selection of attractive, useable and applicable components non-trivial. Comparable research in this area is not known. A further reason that makes the establishment of global identity management systems difficult is the complexity of infrastructure aspects, as

- often every element has some relation to every other element of an infrastructure and
- identity management infrastructures must be interoperable among themselves or with existing legacy solutions.

This part of the book cannot cover all aspects of infrastructure and infrastructure research, but concentrates on three most relevant aspects:

1. Privacy for service oriented architectures (Chapter 21): How can privacy be integrated into service oriented architectures, that define more and more aspects of internet-based business?
2. Privacy and Identity Management on Mobile Devices (Chapter 22): Emerging Technologies and Future Directions for Innovation.
3. Privacy by sustainable identity management enablers (Chapter 22.9): To optimise sustainability, an economic valuation approach for telco-based identity management enablers is presented.

Together these chapters address the roles that networks, or network architectures, devices, and services play for infrastructures considering the interests of the respective stakeholders.

## Chapter 21

# Privacy for Service Oriented Architectures

Ulrich Pinsdorf, Laurent Bussard, Sebastian Meissner, Jan Schallaböck, and Stuart Short

**Abstract** This chapter describes requirements for privacy in service-oriented architectures. It collects 39 legal and technical requirements, grouped in the five categories. These requirements are the starting point for a technical framework that brings privacy-enhanced data handling to multi-layered, multi-domain service compositions. We describe an abstract framework that is technology agnostic and allows for late adoption also in already existing SOA applications. We describe the general building blocks that are necessary on a PII provider's side and on a PII consumer's side. Finally, we look at the technical implementation of a very common, yet complicated aspect: the composition of policies when composing information artifacts. We describe how the composition of data influences the composition of policies.

### 21.1 Introduction

SOA is a technology-independent architecture concept adhering to the principle of service-orientation. It aims at enabling the development and usage of applications that are built by combining autonomous, interoperable, discoverable, and potentially reusable services. These services jointly fulfill a higher-level operation through communication. They fall into the class of distributed systems [CDK05].

One core principle of SOA is the so-called loose coupling of partial services: single services are not permanently bound to each other, rather their binding happens only at run-time, enabling a dynamic composition of services [CK05]. Moreover, it is even feasible to dynamically bind services hosted in different security domains and by different legal entities. We refer to this as “cross-domain service composition” [BNP09]. One prominent example of this are services provided via so-called “service chains” that comprise several partial services offered by different organisations. To facilitate the use of such services, usually one legal entity might serve as single point of contact for (potential) customers. Currently, in the Internet era, the

locations of organisations providing partial services for one high-level service can be widely distributed around the globe.

In many cases, an SOA might involve the processing of personal data and thus pose risks for the privacy of the data subjects concerned. Two specific risks can be identified with regard to cross-domain service composition. The first concerns the lack of transparency with regard to processing personal data: the involvement of different legal entities may lead to the situation where data subjects are no longer aware of what data are handled by what entity for what purpose. Data is exchanged between service providers and the user can only guess which data goes where. This is particularly true if services are invoked dynamically at run-time. In case a high-level service delivery involves different organisations, but is exposed by only one of them, customers might not even be aware of the involvement of further legal entities at all.

The second risk concerns the issue of data linkability: The use of standardised formats and interfaces within an SOA facilitates the linkage of systems and data sets. Since SOA method calls transport typed and semantically well-defined data, it is easy to use this meta-information to link the transmitted data with other data sets. Without the implementation of the appropriate technical and organisational measures, organisations could be able to link different sets of personal data and generate profiles on data subjects.

However, the implementation of an SOA also provides some options to achieve a high level of privacy for the data subjects concerned. First, each single service that forms part of an SOA usually serves a specific purpose such as authentication or payment. In combination with privacy-compliant logging techniques, this circumstance can be used to implement an automated review of adherence to the privacy principle of purpose limitation. Second, tailoring single services to specific purposes simplifies the determination of personal data that are really needed for the implementation of the respective service. This circumstance facilitates adherence to the privacy principles of collection, use, and disclosure limitation as well as obedience to the principle of data minimisation. Third, an SOA provides possibilities for the implementation of an automated data protection management. This results from the fact that, nowadays, technical integration of an SOA typically takes place on the basis of web services and XML. As the same holds true for existing and emerging standards for an automated data protection management (e.g., P3P, EPAL, XrML), these standards could easily be applied within a SOA.

As in Part IV and especially in Chapter 17, we follow the principle of “downstream” data usage. A entire application is modularised into services. Services are specialised in a certain part of an overall workflow and invoke other services as necessary. This principle usually leads to a chain (or even a tree or graph) of service invocations. In terms of privacy, this implies that a service provider whose service is a downstream part (those that process data later) of the overall workflow must adhere to policies given by service providers whose services are upstream parts (those that process data first) of the workflow (cf. Requirement 27 on page 390).

In Section 21.2, we summarise 39 requirements for privacy in SOAs, grouped into five categories. These requirements set the scope for the abstract privacy frame-

work, which we describe in Section 21.3. Finally Section 21.4, finally, looks at the technical implementation of a very common, yet complicated aspect: the composition of policies when composing information artifacts. We conclude this chapter with an outlook and some thoughts on open issues.

## 21.2 Requirements for Privacy in SOA

Service-Oriented Architectures expose new chances and challenges for privacy and data protection. The potentially increased distribution of personal data across multiple domains makes *subject access requests* difficult to handle. Which service processed what data? Whom to address for liability issues? At the same time, service orientation offers a new approach for the granularity of data processing, allowing clearer responsibilities and better auditing.

This section describes a comprehensive set of requirements for Service-Oriented Architectures. If the requirements are applied in the construction of Service-Oriented Architectures, legal compliance with privacy legislation is facilitated. Moreover, the requirements may provide guidance for the design of privacy enhancing Service-Oriented Architectures. They include the privacy risks and opportunities resulting from the implementation of Service-Oriented Architectures within one organisation, but also across different organisations (cross-domain service composition).

In particular, such cross-domain service composition involves new privacy risks. The involvement of different legal entities may lead to the situation where data subjects are no longer aware of what data relating to them are being handled, which entity is doing so and for what purpose. Furthermore, the use of standardised formats and interfaces across different security domains facilitates the linkage of data sets and thus allows for profiling of data subjects.

On the other hand, SOAs can also provide several options to improve privacy and data protection. First, one typical property of any SOA is that each single service could be mapped to specific purposes. This circumstance facilitates the implementation of an automated review of adherence to the privacy principle of purpose limitation. Second, the tailoring of single services to specific purposes also simplifies the determination of personal data that are really needed for the respective service. It thus eases adherence to the privacy principles of collection, use, and disclosure limitations as well as compliance with the principle of data minimisation. Third, as the technical integration of an SOA typically takes place on the basis of web services and XML, it provides some possibilities for the implementation of an automated data protection management.

For reasons of brevity the requirements in this subsection are laid out with little or no examples given. To better understand the concepts described please cf. [MS09].

### ***21.2.1 Core Policy Requirements***

Policies are used by service providers to describe restrictions on the processing of personal data. From a privacy point of view, policies on purpose limitation, non-disclosure and data retention period are of major importance.

**No. 1:** *Policies should be available in an unambiguous formalisation. Thereby, the content of policies should be machine interpretable.*

Since policies should be available for automatic processing and comparison with user preferences, they have to be available in a machine-interpretable form. To avoid misinterpretation of policies and thus reduce legal conflicts, unambiguity in the formalisation is necessary.

**No. 2:** *It must be ensured that communicated policies cannot be disputed by the ensuring entity.*

Policies must be binding, i.e., the ensuring entity must not be able to dispute their existence and exact content.

**No. 3:** *Policies must be easily accessible to users. Accessing the policies should be determined by a clear specification.*

Potential users of a service should be able to see the policies of every service provider without trouble. A standardised means of access could be made available, but should only require a minimum of personal information about the user exercising his/her right of access.

**No. 4:** *Policies should be presented to users in an easily comprehensible manner.*

As policies can be very complex, users that do not have detailed legal knowledge might not be able to understand and assess them. Thus, policies should be described in a manner that is easily comprehensible to the general public. Hereby, the principle of transparency is put into effect.

**No. 5:** *It must be explicitly specified who is responsible for the policy, including a reference to the applicable jurisdiction. This specification must be visible to users.*

The specification of responsibility fosters the principle of accountability.

**No. 6:** *It must be explicitly specified what data are covered by a policy. This specification must be visible to users.*

A clear link between data and policy is needed, since different services of one provider may give varying policies respectively for different parts of one set of data. It is advisable to communicate policies for each purpose separately. Thus, the principle of purpose limitation is facilitated.

**No. 7:** *Policies should cover all aspects of data processing with regard to privacy legislation.*

Policies can be more or less detailed. In order to prevent unnecessary complexity, they should not be more detailed than legally/contractually necessary. Taking a layered approach may additionally foster the principle of transparency.

**No. 8:** *Recipients or categories of recipients to which the data will be passed on to, must be explicitly specified. This must include a reference to the applicable jurisdiction for the recipient.*

The specification of (categories of) recipients fosters the principle of transparency.

**No. 9:** *It should be explicitly specified under what policies data is passed on to other parties.*

If personal information is passed down a service chain, the receiving service provider is legally bound with regard to what it may do with this data. As this may be different from what the originating service may do, it should be reflected in a separate policy.

### **21.2.2 Privacy Logging Requirements**

**No. 10:** *Logging data should be unambiguously formalised and represented in a machine interpretable format.*

If logging takes place in a log file jointly used by different organisations that form part of a cross-domain service composition, a common log format is to be specified. Even if each service provider generates its own log files, the use of a standardised log format facilitates partly automated access to logging data.

**No. 11:** *It must be possible to check the compliance of processing operations with communicated policies on the basis of log files afterwards.*

Using log files allows for the reconstruction of data processing by the service. Thus, it is possible to match policies and log files and to identify incidents that are not compliant with the policies. Adherence to this requirement brings into effect the principle of accountability.

**No. 12:** *It must be ensured that log files cannot be contested by their originating entity in charge of the processing.*

Not only policies (Requirement 2), but also log files must be binding. The originator must not be able to dispute that it generated the log file in its existing form.

**No. 13:** *The fact that data are logged must be visible to the user.*

When the processing of personal information is logged, the logs will most likely contain personal information as well. The user must be informed of the fact that logging is applied, and information on the specific logs may be in scope for subject access requests.

**No. 14:** *The originator of a logging entry must be clearly visible. In particular, it must be visible which service provider of a cross-domain service composition is the originator of a certain logging entry.*

One purpose of logging is proving the legality of the data processing. It must therefore be clear which entity created a log entry. This is especially relevant if several entities write to the same log file.

**No. 15:** *A simple methodology must enable the user to access logging data that s/he has a legal right to access, or that the service provider wants to grant access to.*

The user's right of access includes the right to know what data have been processed for what purpose and whether they were changed. In some cases, the service provider might be interested in allowing access beyond what is needed for legal compliance to support the trust relationship with the user.

**No. 16:** *It must be clear to which data a log entry refers.*

Logs are one source of information for subject access requests. For this purpose, logs must describe actions that were applied to personal information (such as modifications, transmissions, possibly also simple reading access). These actions could be applied to different kinds of data. Therefore the log must be unambiguous in describing to which data it refers.

**No. 17:** *Log files should describe all contractual and further legally relevant aspects of data processing. Beyond that, technical aspects should only be described in case they are relevant.*

Obviously logs can become extensive and large amounts of data can be produced. Not all actions, however, need to be logged, but only those that are relevant with regard to data protection (in particular the right of access). Most of the actions applied on the data, especially changes, corrections or deletions.

**No. 18:** *Log files must contain explicit information on recipients or categories of recipients data have been passed on to. This includes a reference to the applicable jurisdiction.*

This requirement is derived from the legal duty to ensure transparency with regard to recipients or categories of recipients of personal information.

### 21.2.3 Requirements for Access to Personal Information

**No. 19:** *Access to personal information should be provided in an unambiguous formalisation. The content of the information should be machine interpretable.*

Unambiguousness of formalisation supports the correct interpretation of accessed information and prevents possible legal disputes about differently interpreted information. A machine interpretable formalisation enables users of a service to analyse accessed data in a partly automated manner.

**No. 20:** *It must be ensured that access to information that has been granted cannot be disputed by the granting entity.*

The response to subject access requests must be binding. The granting entity must not be able to dispute the information it has communicated.

**No. 21:** *A simple methodology with regard to request and granting of access to information should be provided to users.*

A standardised procedure for the granting of access should be used in order to keep efforts for such access low on both sides. Through standardised clauses a - partial - automation of the process could be feasible.

**No. 22:** *Users accessing information must be enabled to easily recognise what data covered by what policy have been disclosed to what third parties.*

If the personal information of users is processed when a service is invoked, they have the right to obtain information from the service provider about categories of processed data, purposes of the processing, and recipients or categories of recipients.

**No. 23:** *Accessed information should cover only contractual or further legally relevant aspects of data processing.*

Service providers are legally obliged to grant users access to specific information (see Requirement 17). In principle, the accessible information should be limited to this specific information in order to avoid too much complexity. This serves the principle of transparency.

**No. 24:** *Users must be enabled to access explicit information on recipients or categories of recipients that data have been passed on to. This includes a reference to the applicable jurisdiction.*

### 21.2.4 Cross-Domain-Specific Requirements

**No. 25:** *It must be possible to maintain communicated policies even if the Service-Oriented Architecture is dynamically adapted (refers to the constellation of an SOA being established by several entities).*



It may happen that a member of a Service-oriented architecture leaves the organisation and is replaced by another entity. Dynamic changes of this kind should be possible without resulting in the need to negotiate policies once again with customers or even in the necessity to terminate contracts with customers.

**No. 26:** *If it is not possible to maintain (all) communicated policies in case of an adaptation of the virtual organisation, it must be possible to adapt the communicated policies (builds on Requirement 25) through renegotiation.*

In cases of renegotiation, mechanisms have to be in place allowing for the adaptation of already communicated policies to the new conditions in mutual agreement.

**No. 27:** *A service provider whose service is a downstream part of the overall workflow must adhere to policies given by service providers whose services are upstream parts of the workflow.*

As the service provider who is in contact with the customer makes binding policies for the entire workflow, service providers whose services are downstream parts of the overall workflow have to adhere to these policies.

**No. 28:** *Multi-level-matching within a Service-Oriented Architecture must be supported.*

Multi-level-matching takes place when a Service A, which is invoked by a user, launches another Service B. In this case, Service A has to integrate the policies of Service B.

**No. 29:** *The ability of the data subject to have access to information must be ensured for the future.*

If a service composition or a virtual organisation is decoupled, it may be difficult to identify all parties that participated in the specific service afterwards. Therefore, mechanisms are to be implemented that allow subject access requests even in such a case.

**No. 30:** *An ex post notice must be enabled by the appropriate mechanisms.*

If policies change, the user is to be informed subsequently (see Requirements 25 & 26). Therefore, mechanisms need to be implemented that allow for notice in multi-level workflows, even if the user is not known to all services. Equally, it must be possible for the user to accept the changes towards all included services.

### **21.2.5 Requirements for Additional Mechanisms**

**No. 31:** *It must be ensured that the correction and erasure of user data are feasible.*

Data protection legislation gives each data subject the right of rectification and erasure of his/her data to be used towards the controller of the processing. Thus, the service provider must be capable of specifically manipulating personal information about users.

**No. 32:** *It must be ensured that blocking of user data is feasible.*

If data can or may not be erased, there must be a mechanism in place that restricts their further use to the necessary minimum for the given situation. Partial blocking of subsets of a larger data set must be feasible.

**No. 33:** *It should be made easy for users to exercise their rights of correction, erasure and blocking.*

As correction, erasure and blocking are instances that are initiated by the user, technical feasibility as such (Requirement 31) is not sufficient, Rather, it shall be smoothly possible for the user to exercise his rights.

**No. 34:** *It should be possible to guarantee compliance with communicated policies.*

For this, a mechanism is needed that technically prevents the service provider from infringing its policies – i.e., that a part of the provider's infrastructure must be exempted from the provider's direct control.

**No. 35:** *There should be a possibility to support trust between user and service provider.*

There is a need for an infrastructure that enables users to come to trust an unknown provider. This can be built through reputation, amongst other mechanisms.

**No. 36:** *The user shall have the possibility to express his/her preferences in an easy manner.*

As users are quite often technical and legal amateurs, tools enabling them to express their preferences in a formalised manner (as defined by Requirement 1) should be made available to them. These tools should be easy to use. The preferences can be the basis of a partly automated negotiation of new policies.

**No. 37:** *User and service provider should be able to match preferences and related policies.*

In principle, a match of preferences and policies could be processed either on the service or on the user side. To allow for different market requirements, such a matching should be possible on both sides. If both parties are enabled to do the matching themselves, a manipulation by one party would become obvious to the other.

**No. 38:** *Matching of preferences and policies must be comprehensible.*

Matching must take place in such a way that both the user and the provider can comprehend the result, and that the entity processing the matching can reason, that it was done correctly. In those cases, where preferences and policies do not match, negotiation mechanisms could be employed that adopt the policies according to the preferences or vice versa.

**No. 39:** *Mechanisms to express the anonymity set with regard to a specific data type should be supported.*

Every operation should make a statement about the influence that its functionality has on the anonymity of a given set of data records. This allows for estimating the overall level of anonymity that a workflow provides.

### 21.3 Abstract Framework Addressing the Lifecycle of Privacy Policies in SOAs

This section gives a generalised overview of privacy-friendly data handling in cross-domain service compositions. We sketch a generic framework that describes the general processing steps to achieve privacy compliance and proper data handling. The framework is designed in a way that it addresses multi-step data sharing by the repeated application of the same principle. Hence, we use the SOA design principle idea to chain services by chaining our protocol to archive proper data handling in a multi-domain service composition. The framework abstracts from concrete technologies and policy languages and is thus intended to support implementation based on arbitrary technologies for Service-Oriented Architectures. This approach allows for both an abstract consideration of privacy implication and a late adoption. Late adoption means that is any existing SOA application shall be able to be enhanced even after deployment. This may, of course, cause invasive correction in the current implementation of the services, deployment of new components, and mutual agreement of data provider and data consumer. Finally, this framework focuses on the lifecycle of privacy policies and can be complemented by other privacy-enhancing technologies that are described throughout this book, such as data encryption, anonymous credentials, anonymous communication, trustworthy user interfaces etc.

We introduce the idea for an abstract policy framework with a description of the simplest possible scenario, a client-server interaction. Client-server technology can be seen as the ancestor of Service-Oriented Architectures, yet it is still the nucleus of each service-oriented architecture pattern, since even the most complicated service interaction can be broken down to individual interactions between two entities, which represent a client and a server.

Figure 21.1a shows the usual scenario for privacy policies, which is also applicable to client/server systems. The service exposes a privacy policy (e.g., P3P [W3C06]) expressing how it will handle collected data. The user has privacy preferences (e.g., APPEL [W3C02]) that reflects her expectations in terms of privacy. By

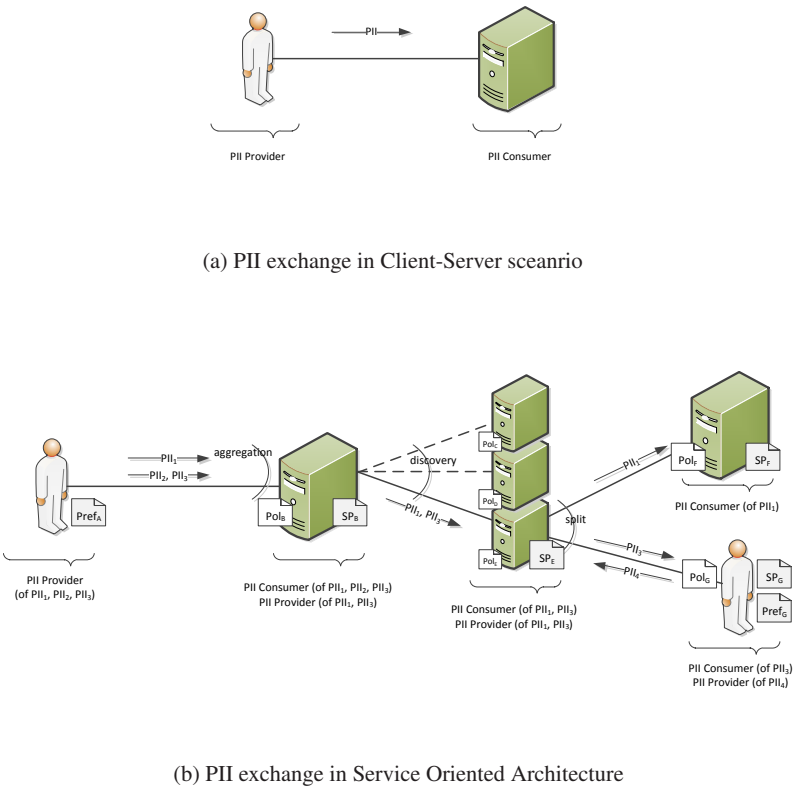


Fig. 21.1: Comparison of privacy policies in client/server architectures and in Service-Oriented Architectures.

comparing privacy preferences and privacy policies, the user (or user agent) determines whether it is suitable to share data.

Service-Oriented Architectures can be seen as applying client/server communication in a recursive way. The user invokes a single service. The service does not perform the full operation itself, but invokes one or many other services to perform parts of the task. These invoked services in turn invoke others services. The result is a tree<sup>1</sup> of service invocation where each node represents a service. We can apply the same recursive pattern to privacy policies that are communicated between the individual services in an SOA.

<sup>1</sup> In theory, the service invocation would even form a directed graph containing loops. But for sake of simplicity, and based on the common practice, we assume the invocation graph is a tree.

Data is collected by a service (data controller) that may share it with third parties. When third parties act on behalf of the data controller, they are referred as *data processors* (cf. 16.1.1). When third parties are in a different trust domain, they are referred as *downstream data controllers* (cf. Section 17.1). In the latter case, the policy of the third party is taken into account when deciding whether data can be shared.

### 21.3.1 Privacy Issues Arising from SOA

Figure 21.1b shows the complexity added by a Service Oriented Architecture (SOA), where different parts of the service are offered in different trust domains. The extension from client/server to SOA adds a number of problems to the scenario with regards to privacy.

*PII provider* is the role of entities sharing personal data with PII consumers. In most scenarios, the user (or data subject) is acting as PII provider. Sharing personal data with another party is generally restricted by privacy constraints (access control and/or expected usage control). Those privacy constraints can be locally specified (e.g., a data subject can specify privacy constraints on her data), can be external i.e., provided by another party (e.g., a data controller sharing collected data with a third party has to enforce constraints imposed by the data subject), or can be a combination of local and external constraints.

*PII consumer* is the role of entities collecting personal data provided by PII Providers. In most scenarios, a service (or data controller) is acting as PII consumer. PII consumer is in charge of enforcing agreed usage control on collected data. Usage control is imposed by the PII Provider and can be refined by the PII consumer.

It is important that, unlike in the simple client/server setting, all services that are neither the root nor the leafs of the “invocation tree” may switch their roles during the process. For example, if a service typically receives a call in the role of a PII consumer, and the parameters of the call bear personal data, the service can switch to the role of a PII provider as soon as they invoke another service. This is based on the assumption that this second call (as PII provider) forwards part of the personal data that the calling service has just received.

PII provider and PII consumer in a downstream scenario are not necessarily only machines. Figure 21.1a illustrates that users share data with services. However, a human being can also collect personal data and become a PII consumer. In this case, either this person has a privacy policy expressing how he/she handles collected data or sticky policies are shipped with the data. This is comparable to a license in rights management.

Other issues arise from the distributed nature of an SOA. First of all, we have to assume that each service is part of a different trust domain. We assume that each service behaves as intended to an upstream service, i.e., we have no malicious behaviour of the services. This is a common trust assumption that is enforced by reputation, audit, certification, and/or trusted hardware and software.

One principle of SOAs is the late binding of services. That means that the concrete instance of an invoked service can be retrieved only at invocation time. This allows for the selection of a service that provides a certain service level.

We can apply this principle in privacy-aware SOAs as well. A data provider may choose the data consumer from a number of data consumers based on the privacy policy of this service. We call this *privacy-aware service discovery*. It is interesting that privacy, when regarded as one attribute in a Service Level Agreement (SLA), competes with other SLA attributes such as price or quality of service. As a result, a service with a more user-friendly policy would appear more often or would rank higher during discovery. This may lead to competitive incentive to provide suitable privacy.

The enforcement of usage control (including access control when sharing data) is done by each party gaining access to a piece of data. This distributed enforcement works properly only when all involved parties adhere to the protection of the data. One way to verify that the data handling was done in the correct way, is to compare the promised behaviour (sum of all sticky policies) with the executed actions, e.g., the log files of all policy enforcement points (PEP). This could be done by a trusted third party. This mechanism could even be federated among parties. That is, each party provides data for promised behaviour and executed actions.

In summary, chaining services adds privacy issues to PII handling that goes way beyond the relatively simple data sharing model in client-server settings. It is very difficult for the user to understand which PII goes where and why. In most cases the user is not even aware that PII he/she discloses with one service is shared with a third party. Moreover, due to dynamic binding, these third parties may only be known at the time of invocation. This situation does not satisfy the requirements we gave in Section 21.2 at all. Hence, we propose a slightly modified communication protocol between SOA entities and an internal service workflows that help to fulfil the privacy requirements.

### 21.3.2 Abstract Protocol

We will now describe an abstract protocol that takes into account the complexity issues for privacy that arise from SOAs. The protocol is technology agnostic, which has the advantage that it may be implemented in various ways. It is a blueprint for how to build privacy-enhancing SOA applications or – in case of late adoption – a blueprint for how to make a SOA application more privacy-preserving.

Figure 21.2 shows the abstract framework from a high-level perspective. It shows a PII provider on the left and a PII consumer on the right. This interaction can be applied recursively on all segments in a service chain of an SOA application. A PII consumer will then take the role of a PII provider as explained earlier, when it invokes other services.

The PII provider side consists of three building blocks: the PII provider behaviour, the PII store and the preferences store. The structure of the PII consumer

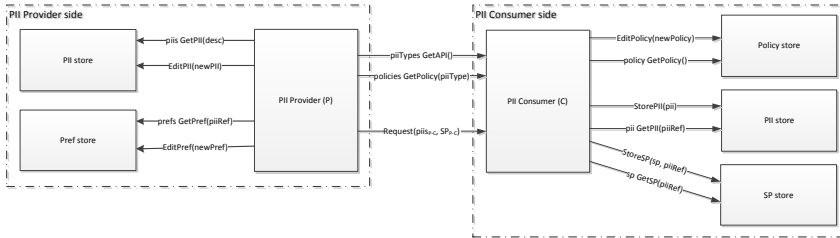


Fig. 21.2: Overview of the generic privacy lifecycle in SOA applications.

side matches and extends the structure of the PII provider, which is an enabler for switching the roles from PII consumer to PII provider. The consumer part consists of the PII consumer protocol, the PII store, the policy store (which is comparable to the preference store), and an additional sticky policy store.

The *PII Store* is the database for storing personal data. Depending on scenarios, this can be a local database (e.g., on client machine), a service in the same trust domain (SQL server at data collector side), or a service offered by a trusted third party (e.g., cloud storage), a local credential store (certified PII), a remote credential store (e.g., Security Token Service), or a combination of them. The *Preferences Store* stores all privacy constraints. The Preferences Store keeps track of constraints that are locally defined (and can be overwritten) and committed (and must be enforced). The *Policy Store* contains the information necessary to describe how data will be handled when sent to the PII consumer. A policy can be statically defined or derived from business processes. A policy can be generic or specific to a user (i.e., depending on authentication). Moreover, a policy can be local or can depend on external policies (e.g., the policies of downstream services). The *SP Store* (sticky policies store) stores the policies that were agreed upon between the PII provider and the consumer for a specific piece of information. The sticky policy is a result from matching<sup>2</sup> the PII provider's preferences with the PII consumer's policy.

PII Provider and PII consumer communicate via a simple three-step protocol.

1. Requested PII types. First, the PII provider asks the consumer side for the PII types that are needed for the service invocation. This request may have different technical incarnations. It could be a specification in a web form, it could be a service description such as WSDL<sup>3</sup>, but it could also be a dedicated method call to request this meta-data about the service that the PII provider intends to access. In any case, the PII provider learns about the types of information that are requested to perform this service.
2. Policy request. In the second step, the PII provider asks for the policy that shall be applicable to the provided information. In other words, it requests a descrip-

<sup>2</sup> Policy matching is defined in Chapter 17.

<sup>3</sup> WSDL stands for web service description language.

tion (in a formal language) of the PII consumer's commitment on how collected data will be handled.

3. Service invocation. In the third step, the PII provider submits the requested PII together with a sticky policy for each data item.

Internally, this three-step protocol is backed up by many sub-procedures and decisions. We will look into this individually for the PII provider and the PII consumer. From this high-level perspective, it is important to understand that the PII store provides the personal data that is shared in the third protocol step. The preference store allows the retrieval and editing of stored information. The preference store participates in the creation of a sticky policy for the submitted data items. The privacy policy provided by the PII provider side is compared with the preferences for this specific piece of information. The result of this matching process is a minimal policy that considers both the PII provider's preference and the PII consumer's policy. Please refer to Chapter 17 for more details on policy matching and the creation of a sticky policy. For now, it is enough to consider the sticky policy as the least common denominator between privacy preferences and a privacy policy.

The internal components on the PII consumers's side look very similar to the setting at the PII provider's side. The policy store allows for retrieving and editing policies, which are sent in the second protocol step. The PII store is needed to store the received PII from the PII provider. While the PII is kept in the PII store, the sticky policy is stored in the sticky policy store. However, a linking mechanism, e.g., a dedicated link table in database, keeps a relation between the piece of PII data and the sticky policy.

This high-level protocol could be embedded in an existing application interaction. The last protocol step is usually the service invocation of any given SOA application. All we do is add an interaction step that requests the required information before the actual invocation. This is nothing new. Service discovery mechanisms provide this information anyway, but usually only on a data type level. For instance, a WSDL description clearly states that a function call `calculatePension(date)` requires the parameter to be a data type encoding a date. Currently, this service description is usually on a syntactical level. In other words, the service description does not state that the submitted data has to be the user's date of birth. However, semantic web technologies do specify this. Mechanisms for requesting policies are widely used for Service Level Agreements (SLA). Hence, both protocol steps 1 and 2 are special ways of meta-data lookups. We request that the service does this privacy-specific meta-data lookup before the real service invocation.

Figure 21.3 shows the Abstract Privacy Framework in more detail. The figure still contains the top-level components for the PII Provider and the PII consumer (dashed boxes). Furthermore, it shows the iterative approach by chaining from the PII consumer to another PII consumer. Each top-level component contains a best-practice workflow. This workflow is an ideal, scenario-independent view on data handling in a composed service. Moreover, the workflows introduce more components that should be part of each role's technical representation. The next sections describe all top-level components from Figure 21.3.



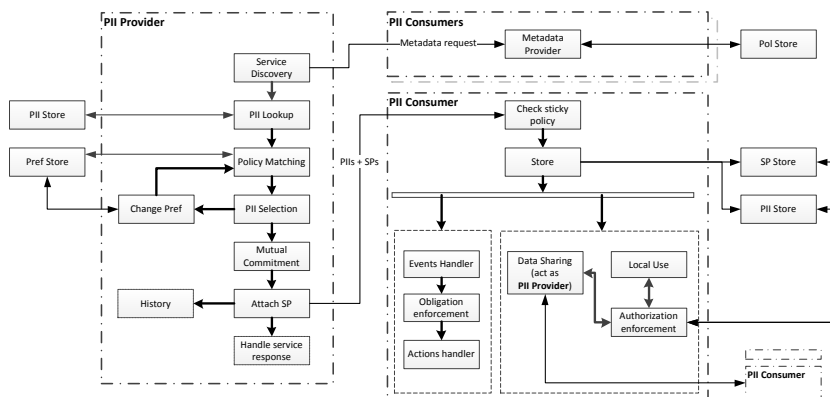


Fig. 21.3: Generic privacy lifecycle in SOA applications in detail.

### 21.3.3 PII Provider

*PII Provider P* is the role of entities sharing personal data with PII consumers. In most scenarios, the user (or data subject) acts as PII provider. Sharing personal data with another party is generally restricted by privacy constraints (access control and/or expected data handling). Those privacy constraints can be locally specified (e.g., a data subject can specify privacy constraints on her data), can be external i.e., provided by another party (e.g., a data controller sharing collected data with a third party has to enforce constraints imposed by the data subject), or can be a combination of local and external constraints. The PII Provider's role is essentially about deciding whether it is worth sharing pieces of personal data in order to obtain services from PII consumers.

This section looks into the box labeled “PII Provider” in Figure 21.3. It illustrates the general steps a PII provider has to undertake internally in order to support a privacy preserving protocol we described in the last section. The figure is intended to be a block diagram, so it is neither just a workflow nor a collection of software components, but rather a mixture of both. The *Service Discovery* step refers to a mechanism to find and select one or more Data Controllers to be used and gain their meta-data, such as functionality, credentials, or SLA. Bringing together PII Provider and PII consumer is necessary before any interaction can take place. We call this phase service discovery even if, in some scenarios, it can be initiated by the PII consumer. In our picture, requesting the required PII and requesting privacy policies is covered in two separate steps, but they could technically be merged into a single meta-data request.

*PII Lookup* is a mechanism to determine whether the PII requirements can be met by the personal data in the PII store. It aims at finding all combinations of personal data that may satisfy the request of the PII consumer. The PII Provider can have

different personal data that match one element of the request and can even load or create new personal data (enter a date in a form, attach a picture) with no a priori attributes. The PII consumer may accept different types of personal data (e.g., name and age), may specify different attributes of PII (e.g., signed by a given third party), and may accept special combinations of personal data (e.g., credit card number and expiry date must come from the same credit card credential).

The *Policy Matching* mechanism decides whether personal data can be shared according to its privacy constraints and the preferences of the PII consumer. The selected PII must not only match the requested PII in type and semantic, but also the privacy preferences associated with the individual PII data items. Privacy preferences settings are always specific to a given personal data. This can be the combination of different privacy preferences. For instance, an e-mail address can be subject to preferences related to any address, to any e-mail address, and to this specific e-mail address. Moreover, a given piece of data can be subject to constraints (preferences and sticky policies) from different parties e.g., data issuer (for a credential), data subject, or data controller (local preferences). The preferences of the selected PII must match with the privacy policy of the service. Chapter 17 describes this process in detail. The PII selection process may be very complicated since the selected PII and its associated preference mutually influence each other and must comply with the service policy. For instance, the user could have multiple credit cards with different privacy preferences. Hence, the user can influence the policy matching process by selecting the specific piece of PII and by changing the policy preferences sticking to this PII. The block diagram reflects this with a loop around PII selection, policy matching, and PII selection.

*PII Selection* is a process which allows the user to pick the appropriate PII from the PII store. PII selection can be seen as an extension of Identity selection where not only minimal disclosure is taken into account but also privacy policies.

The selection could be done automatically or in a manual process by the data controller. It is already a challenge to find a suitable solution in the search space across multiple user credentials, but it may be even more challenging to visualise this complexity to the user. In case of more than one suitable solutions the user must also be empowered to understand what the best solution is. This needs proper user interface support (cf. Section 14 for more details). Depending on the actual case, the user may want to pick the solution that enforces the most restrictive privacy preference, the solution that shares the least amount of data, or the solution which does not use a specific set of credentials.

Thinking about the non-matching case is as interesting as finding a valid combination of credentials with associated preferences. When there is no suitable option, the user needs to understand why there is no match and propose different options to proceed. The user could simply stop the transaction and not invoke the service at all. Probably he/she would repeat the service discovery step and pick another service that provides a similar functionality. The user could also continue the processing and violate his/her own preferences, or adapt the preference in the preferences store so that the PII selection yields at least one match. In principle, the service could

adapt its privacy policy as well, e.g., if the user picks a more expensive service level (“premium service” vs. “free service”).

For the user to decide to amend his/her privacy preference in order to achieve a match with the PII consumer’s policy, the step *change preferences* is taken. Note that different types of updates can be envisioned ranging from adding a preference for this specific case to changing the preferences that covers a larger group of PII consumers and/or personal data.

Finally, when the PII selection was made that matches the policy, we create a *mutual commitment* between the PII provider and the PII consumer. We call this step commitment rather than agreement, since we see a policy negotiation as an optional step. In other words, we assume for most use cases, the privacy policy of the PII consumer is fixed and any adaptation will be made on the preferences of the data provider. Hence, this leads to a mutual commitment, but not necessarily to a negotiation.

The agreement itself can be a sticky policy or just a Boolean response indicating an acceptance of the PII consumer’s policy. A more sophisticated technical implementation could even foresee a statement that is signed by both parties or witnessed by a trusted party.

In case the agreement is expressed with a sticky policy, which we assume here without loss of generality since it is the most expressive form of agreement, the sticky policy may recursively specify usage control that must be enforced by the PII consumers (including downstream). Indeed, usage control may specify access control towards downstream services including downstream usage control.

The action to *attach a sticky policy* involves the communication of the requested PII from the data provider to the data consumer together with the sticky policy. In other words, this step also involves the *service invocation*. When this step has been performed, the PII consumer possesses the requested PII and the sticky policy. Technically, the communication of PII and sticky policy can be performed in a single service call or in two separate calls. Moreover, one sticky policy may apply to multiple pieces of data. In any case, it is necessary to keep the link between the data and its related sticky policy. The PII provider may keep track of this interaction in a history store. This is helpful e.g., when a trusted third party audits the PII consumer at a later point in time, when the user wants to base a PII selection decision on earlier decisions, or when the user wants to verify to whom a certain piece of data was disclosed and under which conditions.

### 21.3.4 PII Consumer

*PII Consumer* is the role of entities collecting personal data provided by PII Providers. In most scenarios, a service (or data controller) acts as PII consumer. PII consumers are in charge of enforcing agreed data handling on collected data. Data handling is imposed by the PII Provider and can be refined by the PII consumer. The PII consumer’s role is essentially about 1) checking whether actions

are authorised before acting on collected personal data and 2) enforcing obligations regarding those data.

This section elaborates the box labeled “PII Consumer” in [Figure 21.3](#). Similar to the previous section, it illustrates the general steps a PII consumer has to undertake internally in order to support the privacy-preserving protocol we described in Section 21.3.2. Again, this figure is intended to be a block diagram, so it is neither just a workflow nor a collection of software components, but rather a mixture of both. The Meta-data Provider is the matching part to the meta-data request on the PII provider side. It provides the requester with a functional description of the service and optionally with the SLA. In this context, it is important that meta-data states information about certification, the PII requirements, and the privacy policy. The information is gathered from the service implementation itself, e.g., it could be part of a WSDL document, and from the policy store. The meta-data can be static, which means each caller gets the same information, or the meta-data can be dynamic, in which case (part of) the information is specific to the context of the request. For instance, different callers (PII providers) may get different privacy policies because the PII consumer shares individual legal agreements with the various PII providers. Another example is that the policy may depend on the geographical region of the caller.

The PII provider digests this information, selects the right PII data, and matches preferences and policies. Finally, the PII provider invokes the PII consumer’s service and thus submits the requested PII and a sticky policy. The PII provider must verify that the sticky policy matches its policy. This check is necessary to avoid service errors or even legal implications through a wrong sticky policy. The sticky policy could, for instance, disregard the PII consumer’s policy and define arbitrary obligations for the PII consumer.

Since we consider sending the PII and the sticky policy as a single step (cf. Section 21.3.3), the PII consumer’s answer is the result of the service invocation.

During or after the execution of the service, the PII consumer may store the data and likewise the sticky policy. Moreover the PII consumer needs to establish a link between both data items. That is, the service provider needs to remember which PII is associated with which sticky policy and vice versa. The PII is stored in the PII store, the sticky policy is kept in the SP store. The link between both goes either as a reference to both stores to allow a bidirectional mapping or is kept in a dedicated data structure that holds references to the respective entries in PII store and SP store.

Even after the service invocation, the PII consumer may want to take advantage of the collected PII and use this data for an allowed purpose. We distinguish two ways of using these data: it might be used for local purposes or it might be passed on to a third party. Local use means that the data is used inside the trust domain of the PII consumer, e.g., by a second service that operates on the same PII store. In this case, the PII consumer has to verify whether or not the data is allowed to be used for the given purpose. A different case is passing on the data to a third party, i.e., a PII consumer in a different trust domain. This is the moment when the PII consumer switches its role to become a PII provider. In both cases, the verification of access rights to stored PII is performed by an access control engine,

which enforces the authorisation. The sticky policy associated with each piece of PII helps the authorisation engine to decide whether the user agreed to this purpose or not.

In addition, guarding collected data whenever it is accessed, the PII consumer has to follow-up on obligations that were agreed upon with the PII provider. It has to react to events (scheduled or relevant) and execute appropriate actions. We foresee two types of infrastructure components in a PII provider-side obligation enforcement engine. First, there are action handlers. The action handlers are the link to the legacy systems in the PII provider's IT infrastructure, e.g., a database or mail server. Their duty is to execute actions on legacy applications such as sending notifications, logging, or deleting data. The second component type is the event handler. It is responsible for handling events from legacy systems that are relevant from a privacy point of view. Typical events are a time-based events and data access events from the legacy system.

An interesting extension is the logging of all obligation enforcement actions and access control decisions. A formalised logging would allow a trusted third party, such as an accredited auditor, to compare these logging data with the obligations in the sticky policy. An automated matching process would enable the auditor to see which obligations were kept by the PII provider and certify the PII consumer accordingly.

### ***21.3.5 Matching Abstract Framework with SOA Requirements***

In this section, we want to compare the abstract framework addressing the lifecycle of privacy policies in an SOA from Section 21.3 with the requirements we sketched in Section 21.2. The framework clearly focuses on the cross-domain-specific requirements (No. 25-30), but it is also a vital infrastructure to achieve core policy requirements (No. 1-9) and requirements for additional mechanisms (No. 31-39). We will give a short reasoning for each of the covered requirements.

**No. 1:** The exchange of machine-readable policies is an essential part of the abstract framework. The proposed infrastructure absolutely rely on the communication of policies between service provider and service consumer.

**No. 2:** One way to achieve binding policies is by electronic signature. The framework does not explicitly enforce such a mechanisms, but the protocol step *Mutual commitment* provides the opportunity for a dedicated commitment protocol.

**No. 3 & 4:** The abstract framework is more of an infrastructure, thus it has no user interaction. But the protocol cycle of *Policy Matching* and *PII Selection* certainly needs to present the policy to the user in human-readable manner.

**No. 5-9:** These requirements need to be addressed on the level of the policy language.

**No. 10-18:** Although logging is not the core purpose of the abstract framework, it foresees a History step on the PII Provider side. On the side of the PII consumer,

logging would be the duty of the action handlers, because the logging should describe what has happened to the PII. Requirements 11-18 specify implementation details of the logging mechanism. This is not explicitly addressed by the abstract framework, but could be fulfilled by the proper selection of technologies.

- No. 19:** Requested PII is described in service meta-data and stored in databases of the PII consumer. This should fulfill the requirement assuming that the semantic meaning of the meta-data is carried on to the *PII Store*.
- No. 20:** Non-disputable access is ensured by sticky policies linked to PII in the *PII Store*.
- No. 21:** The methodology of granting access PII is to attach a sticky policy to the data that fulfills both the PII consumer's policy and the PII Provider's preferences.
- No. 22-24:** These requirements need to be addressed on the level of the policy language. The building block *History* on PII consumer's side helps to achieve No. 22.
- No. 25:** This is one of the main aspects of the abstract framework. It explicitly addresses dynamically adapted SOA. Since sticky policies travel with PII, and PII consumers will change their role to PII Providers when passing on information, the user's intent always travels with the disclosed information.
- No. 26:** The abstract framework does not support the renegotiation of policies, but the sticky policy traveling with the disclosed PII is always matched against the latest version of PII consumer's policy.
- No. 27:** This policy is automatically fulfilled, when the service adheres to the sticky policy communicated along with the data.
- No. 28:** The abstract framework is designed to be used in a recursive way. A former PII consumer switches to the role of a PII provider when passing on data. This ensures multi-level matching. The entire service chain does not need to be known a-priori.
- No. 29:** Informing the user about the whereabouts of his/her data is possible when the user expresses in the sticky policy an obligation in to notify him before passing on the data.
- No. 30:** If a service provider changes its policy, it is relevant only for new service requests. Data received under another policy will be attached with a sticky policy that results from the original version of the policy.
- No. 31-33:** The building blocks *Obligation Enforcement* and the *PII Store* would help to fulfill the correction, erasure, and blocking of data. However, user interaction invoking this right is not part of the framework.
- No. 34:** Compliance is one of the main purposes of the obligation enforcement mechanism on the PII consumer side.
- No. 35:** Trust establishment is not addressed by the abstract framework.
- No. 36:** Privacy preference on the PII Provider side is a core principle of the abstract framework. Again, the user interface aspect is not part of this infrastructure, but *PII Store*, *Policy Matching*, and *PII Selection* require a proper user interface.

**No. 37-38:** Policy matching is again one of the core principles of the abstract framework. It is utilised in the building blocks *Policy Matching* and *Check Sticky Policy*.

**No. 39:** This requirement is not supported. The anonymity set could probably be expressed as meta-data.

Concluding, we state that the abstract framework addressing the lifecycle of privacy policies in an SOA fulfills most of the requirements for privacy in SOAs. Some requirements need to be addressed by a wise selection of technologies instantiating the abstract framework. The choice of the policy language, for instance, has a major influence (No. 5-9, 22-24). Other requirements need to be addressed by means of extra infrastructures, such as establishing trust (No. 35).

## 21.4 Policy Composition

In most cases, the data consumer's privacy policy is shown to the user as natural language text, and the user can accept it or not depending on her privacy preferences. Typically, if the user does not accept it, he/she is not allowed to proceed. This process should be automated to enable more complex interactions to take place, such as supporting the user in tracking the usage of her PII, facilitating the enforcement of her privacy policy and dealing with more intricate cases, where the data exchanged comes from multiple sources, each with its own privacy policy. In the latter case, data must be aggregated as well as the privacy policy. To this extent, particularly relevant is the concept of a sticky policy [APS02]: privacy policies are strictly associated to a piece of data and should be composed whenever data aggregation happens.

Expressing a condition for each piece of data could be a means for data providers to declare how their personal data should be used. For example, it may be specified that the information is not to be transferred to a third party or that it can be used for research purposes only. For example, a job applicant would like to create an electronic CV (eCV) that contains up-to-date information on her personal details (gender, age or race), work experience, academic qualifications and references. This personal data could be entered either by the user or provided by other services such as a university, previous/current employers or an official entity, each of them containing corresponding privacy and access control constraints (e.g., not to be released for commercial purposes or for internal use only), expressed as privacy policies.

The job applicant should be able to compose them in a single document with a single privacy policy, and be able to handle possible conflicts among policies. On the other hand, a job broker service may need to compose job offers from multiple sources and aggregate the corresponding policies. The aggregated policy may simply be the union of the source policies if they express non-conflicting conditions (e.g., one policy defines the purpose as research but it does not impose any condition on a retention period, and the second policy states a specific retention period but it has no requirements on the purpose). In the case where two or more policies



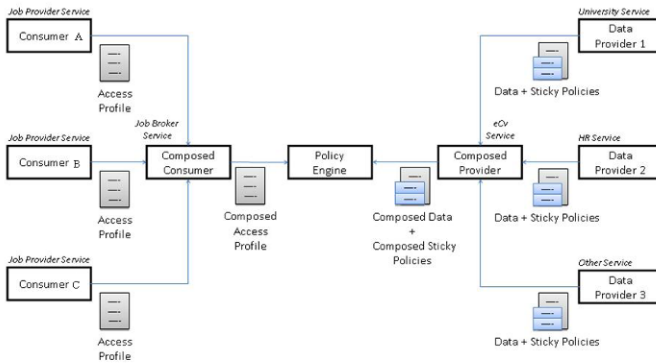


Fig. 21.4: Privacy-aware architecture with policy composition.

specify the same condition (e.g., both state a different retention period), it may still be possible to find an aggregated policy that satisfies both of them or more complex mechanisms of policy compositions would be needed. Even if privacy policy languages exist, such as P3P [W3C06], EPAL [BDS04] or PRIME [PRI], they lack the notion of sticky policies or the complex composition of services or policies for resolving possible conflicts [BZW06].

### 21.4.1 Policy Composition Scenario

To introduce the challenges related to the privacy policy composition, we present an illustrative employment scenario in which users and job providers are able to interact via different web services. A user (job applicant) would like to create an electronic CV (eCV) that contains up-to-date information on his personal details, work experience, and academic qualifications. The personal data, such as the person's gender, age or race, could be entered by the user or provided by an official authority service. Other types of information may include university degrees, recommendation letters and previous or existing employer details, and they could be provided by the corresponding organisation/data provider as a signed digital document or as a reference (Fig.21.4). For example, the university certifies on qualifications attained and a recommendation is usually provided by an academic institution and/or employer.

Each contributory data provider could have a rule or a sticky policy attached to the data that outlines how the data will have to be handled when used by the data pro-



ducer (the job applicant), data consumer (e.g., job broker) or third party. The parts of the eCV that contain this information cannot be altered by the applicant in order to preserve the policy preferences of the different services. These constraints, imposed by such data-providers, may restrict the exposure of some information which is related to a company and should not be revealed. For example, a policy will only allow you to use a recommendation letter for a specific timeframe or it may be the case that the applicant will allow a certain country, such as the United Kingdom, to see his race, as it is a prerequisite to process the application but will not permit other countries to see the information if this is not a precondition.

The final electronic CV is composed of two parts, namely, the composition of data emanating from the different sources and the corresponding aggregated policies. The policy composition may contain conflicts, for example, the applicant may allow his personal contact details to be viewed by all services whereas the company he/she is working for states that it will not permit disclosure of where the employee works for security reasons.

Similarly, on the data consumer side (service side), we have a job broker service that is composed of offerings proposed by job providers or recruiter services. The latter contains rules on data usage conditions; for instance, they will not pass information to other parties, or they will retain the data for a certain period of time. When clients contact the broker with a request for a job offer that corresponds to certain search criteria, the latter selects the appropriate recruiter and puts it in contact with the eCV service. We assume that the job matching is done using the broker's search engine and that the recruiter service is requesting the applicant's CV (so the job broker does not need to know all the details of the CV, but just the search criteria). Subsequently, there will be a policy matching between the two services in a policy engine service (Fig.21.4). When the data provider's privacy policy matches with the constraints outlined by the data consumer then the CV is sent to the job provider service. If this is not the case, then the recruiter's request is rejected and the applicant is notified. Two services should then relax some constraints and try again (policy negotiation).

### ***21.4.2 Privacy Policy Composition Challenges***

The simple scenario described in the previous section outlines some challenges related to the policy composition on the data provider/consumer side. We provide here a non-exhaustive list.

- **Client Vs Server side:** since the privacy policy on the server side is published to announce the way the data will be handled and the policy in the client side is stuck to the private data to describe how this data should be treated, it is important to distinguish between policies emanating from the client side and from the server side. Composing and enforcing such policies should be handled in a separate manner.

Case	Retention	Purpose
$p_a$	*	Research
$p_b$	6 M	*
$p_c$	3 M	Marketing

Table 21.1: Example of the elements of 3 simple possible policies. \* indicate that no condition is expressed, thus all possible values are permitted.

- **Aggregation/Combination:** the privacy policy engine must support the aggregation of policies when multiple policies from different sources refer to a specific piece of data. Aggregation in this context refers to a combination of policies that refer to different data handling constraints not resulting in a conflict. In practice, aggregation is represented by a union of a set of policies. On the other hand, *policy composition* is when the policies refer to the same element and conflict may arise. We will discuss the two cases in more detail below.
- **Trusted third-party:** when the data consumer is composed of different services, then the data provider should be able to deal directly with the relevant trusted third-party (TTP) and not be obliged to divulge information unnecessarily. This TTP filters the relevant data that will be displayed to the server without any indication about the original dataset or the privacy policy.
- **Negotiation:** When the client or the server has a trade-off to make in order to achieve a transaction, it is necessary to find a compromise between conflicting rules. The precedence system should play an important role to automate this requirement. For example, in the context of the scenario, the data producer may be willing to provide PII for research purposes if a criteria such as salary was above a certain threshold.
- **Content-based condition:** Policy conditions may depend on the content of the data, e.g., data may be used for marketing purpose only if age, as reported on the CV, is greater than 21, or job level may be disclosed to third party only if lower than a certain threshold. Such constraints are difficult to address since they introduce an interaction between the data handling policy, which expresses how the data should be used, and the content of the data itself.

To illustrate the problem of aggregation/composition, let us consider a toy-example of combination of two policies with two elements only  $\{\textit{Retention period}, \textit{Purpose}\}$ . Let us call the area in the *policy space* (in our example the  $\{\textit{Retention period}, \textit{Purpose}\}$  space) that fulfills the policy  $p_a$  ( $p_b$ ,  $p_c$ , respectively):  $\mathcal{A}$  ( $\mathcal{B}$ ,  $\mathcal{C}$ , respectively), see Fig. 21.5. For example, policy  $p_a$  is fulfilled for any retention period with the condition to have 'Research' as purpose, policy  $p_b$  for any purpose but with retention of 6 months or less.<sup>4</sup> Let us examine some possible combinations:

<sup>4</sup> We make the assumption here that if the retention period is set 6 months, policy is also respected if it is actually less. This is a reasonable assumption in most use-cases, but there could still be scenarios where regulations impose a minimal retention period. For example, an internet provider may outsource the storage of IP addresses to a third party, defining a corresponding privacy policy that indicates as retention, say, 12 months and not less, in accordance with the regulations.

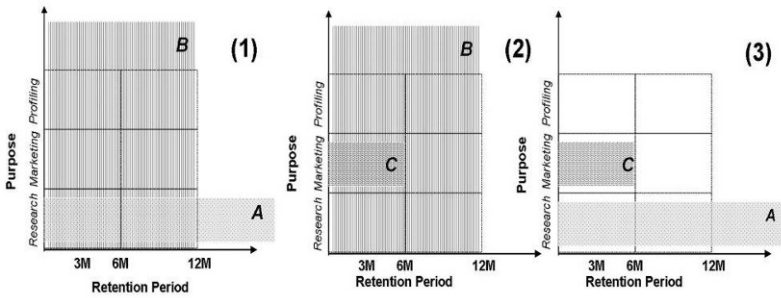


Fig. 21.5: Data Usage: Purpose Vs Retention Period.

- Combining policy  $p_a$  and  $p_b$ . These policies have no overlapping elements, and the aggregate policy is simply the union of them:  $p_a \cup p_b$ , which is fulfilled for the (non empty) area of the policy space:  $\mathcal{A} \cap \mathcal{B}$ , see Fig. 21.5. We call this process: policy aggregation (no conflicts).
- Combining policy  $p_b$  and  $p_c$ . These policies have a common element: retention period. Still, the corresponding areas in the policy space have no empty intersection (under the assumption detailed in the footnote),  $\mathcal{B} \cap \mathcal{C} \neq \emptyset$ , and a combined policy can be easily derived:  $\{\text{Retention period}=3\text{ M}, \text{Purpose}=\text{Marketing}\}$ .
- Combining policy  $p_a$  and  $p_c$ . These policies have conflicting elements (Purpose). Accordingly, there is no policy that fulfills both of them  $\mathcal{A} \cap \mathcal{C} = \emptyset$ . The resulting policy cannot be easily derived, and composition rule should be defined, such as precedence rule (e.g.,  $p_a$  overwrites  $p_c$  because it is coming by an authoritative source), or simply the composition cannot take place.

For policy composition in a general case, rules should be defined. We will not enter in details in this paper; we will only examine a possible case of conflict in the prototype description, which has been addressed introducing a simple precedence rule.

### 21.4.3 Data-Centric Architecture for Privacy Enforcement

In this section, we present a data-centric architecture that addresses the requirements previously mentioned and involves a data flow that follows a path containing services that are able to receive, compose, elaborate, store, and publish data from other services. There are two distinct service types, one being the producer and the other the consumer, with sometimes a service acting as both. A producer stores or produces data while the consumer, who can access all data that is exposed, invokes an operation on the producer to access data that he/she wants to consume (Fig. 21.7).

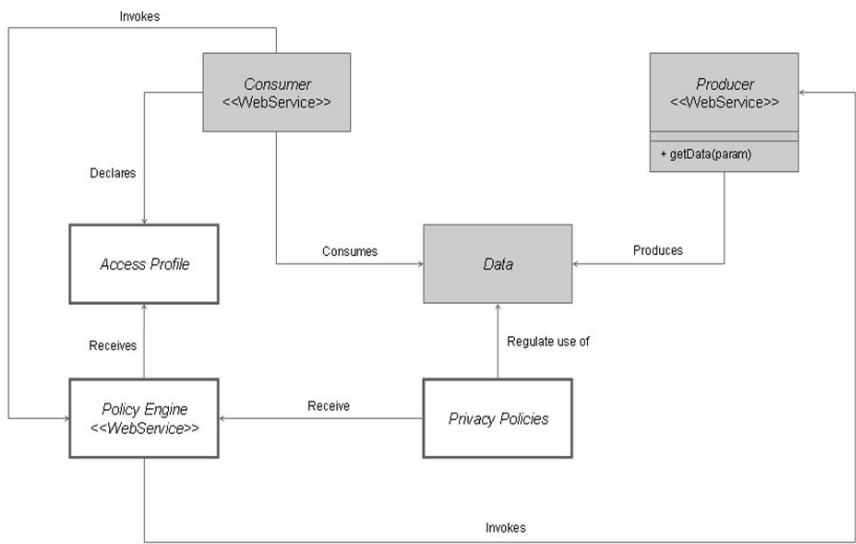


Fig. 21.6: Privacy-aware architecture.

In the context of the eCV scenario, this structure would be a direct link between the eCV and the job broker service, with no policies outlining how the data should be handled between the two services. In order to regulate access to the data, there is a need to add a privacy control from the perspective of both the data producer or provider and data consumer; the former is the eCV while the latter is the job service.

In [Figure 21.6](#), the Access Profile is a document in which the Consumer declares the subject (who it is), resource (which data needs to be accessed) and action (what will be done with the data) as well as information on the usage conditions of the data, such as retention time and disclosure to third party. Privacy policies are a set of rules for accessing the data and are based on the attributes specified in the Access Profile; they travel with the data as sticky policies.

The privacy aware architecture can proceed as follows: the Consumer Service sends an Access Profile request to the Policy Engine; the Engine receives the Access Profile request and translates it into an access query call to the Producer Service; the latter returns data with sticky policies; the Engine compares the Access Profile with the sticky policies in order to enforce the privacy rules over data; depending on the profile of the requester, the Policy Engine will select the parts of data to be displayed or hidden; in the case of a successful matching, the Engine sends back the data to the Requester Service, otherwise there will only be a negative answer.

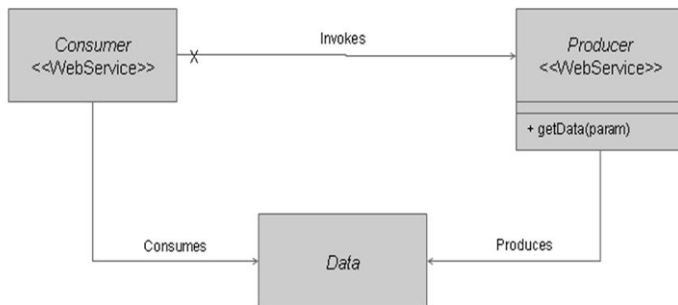


Fig. 21.7: Basic architecture.

When the Access Profile contains data aggregated from multiple sources, the associated policies should also be collected. Since some of these policies can concern common parts of the data, conflicts need to be resolved in order to provide a coherent and secure policy enforcement. In this case, the Composed Provider (eCV service in the scenario) will be in charge of reconciling all of these policies in order to obtain a single composed policy satisfying the privacy constraints related to the sensitive data. The composed policy is then stuck to the dataset and transmitted to the Policy Engine. On the consumer side, in the case of composite service, privacy policies will also be composed and transformed into a single policy by the Composed Consumer (Job Broker in the scenario). The Policy Engine will be in charge of matching the consumer’s policies with the provider’s preferences. After selecting the appropriate consumers, the Policy Engine will enforce the provider’s policy and provide the data that is permitted to be sent, to the consumers.

### 21.4.4 Conclusion

In a situation where both the data consumer and data provider compose their own policies on data usage, it may be necessary to deal with conflicts. This can become more complex when a web service is composed of different services and consequently many policies.

In this section, we looked at privacy policy composition from the perspective of the data provider by proposing a scenario that provides an overview of the issues involved and subsequently outlined a data-centric architecture that could be used to resolve them.

## 21.5 Outlook and Open Issues

This chapter started with requirements for privacy in Service-Oriented Architectures. It collected 39 legal and technical requirements, grouped into five categories. These requirements were the starting point for a technical framework that brings privacy-enhanced data handling to multi-layered, multi-domain service compositions. We describe an abstract framework that is technology agnostic and allows for late adoption in existing SOA applications. The adoption could be partial, i.e., between just two entities of a larger SOA application. We described the general building blocks that are necessary on the PII provider's side and on the PII consumer's side. Finally, we looked at the technical implementation of a very common yet complicated aspect. That is the composition of policies when composing information artifacts. We described how the composition of data influences the composition of policies.

Certainly, this work is far from complete and leaves many open ends for future work. One of the most interesting points is to manifest the abstract privacy framework with a semi-formal notation. This would allow for specifying the protocol and the relationship between the data items in much more detail. It would even allow for describing precisely the complicated relationship between credentials, requested PII, preferences, and policies during the PII selection process. The visualisation of the PII selection is an important point as well that is only partially addressed so far (cf. Chapter 14). Another interesting route is to map existing technology with the abstract framework. It would be interesting to see how existing technology already covers part of the picture. This gives the reader a better method for making a technology choice for his/her SOA.



## Chapter 22

# Privacy and Identity Management on Mobile Devices: Emerging Technologies and Future Directions for Innovation

M. Bergfeld and S. Spitz

**Abstract** Secure Elements have been around as identity providing modules in Mobile Services since the creation of the Mobile Phone Industry. With an increasingly dynamic environment of Mobile Services and multiple Mobile Devices, however, and with an ever changing ecosystem, characterized by new value chain entrants, new (partial) identities need to be provided for the end users. Here, emerging Secure Elements such as Stickers and Secure SD cards can be leveraged in addition to the omnipresent SIM card / UICC. For future services though, even more flexible, secure and privacy enhanced Secure Elements, such as Trusted Execution Environments can be expected. They are needed to cope with an ever more dynamic Mobile Services environment that depends upon reliable, partial identities of the end users and increasingly calls for privacy and security measures. This chapter elaborates upon the emerging and future Secure Element technologies for Mobile Devices. These technologies shall allow an increasingly dynamic creation of services between front-end Mobile Devices and back-end Servers. The Chapter sets the current developments of the ecosystem for Mobile Services into perspective with the needed technologies, reflects on the contributions of the PrimeLife project and draws attention towards the still needed future directions of innovation.

### 22.1 The Status: Privacy and Identity Management on Smart Mobile Devices

With the lifestyle of “Digital Natives” [BT10] spreading throughout the younger generations and the “Wisdom of the Crowd” [BT10] gaining relevance in the creative activity of private as well as professional collaboration, open technology systems that span across numerous individuals and groups are increasingly important.

While these systems empower the dynamic creation of new services and business models, especially through the usage of Mobile Devices such as Mobile Phones, Netbooks, Tablet PCs or even Cars (e.g., via their onboard computers), and their in-



teraction with back-end Servers in Service-Oriented Architectures (SOA), all these opportunities for open collaboration and the rapid sharing of data, information and knowledge also increase the challenges for security, privacy and identity management. For example:

- Providing trusted platforms and security for the execution of services between Front-end Mobile Devices and Back-end servers. Also, securing these interactions against attacks and interventions.
- Providing the possibility for multiple (partial) identities in one Mobile Device, i.e., allowing the end user to consciously manage different identities for different audiences.
- Providing dedicated channels of communication, storage and interaction for partial identities of individuals and assuring that these channels respect the privacy of the individual end user by only making the information transmitted accessible to the respectively intended recipient.
- Providing solutions of anonymity where applicable without jeopardising authentication.

This chapter elaborates upon the emerging technologies for Mobile Devices that are expected to allow an increasingly dynamic creation of services between front-end Mobile Devices and back-end Servers, including the use of technologies that promote security, allow identity management and enhance privacy.

## **22.2 The Changing Context (I): Multiple Partial Identities across Devices**

It has been pointed out that scalable privacy models in conjunction with different levels of security are important for identity management (see [Pri08a]). Most of the underlying technologies such as cryptography are available (see [SSP08]), although they might have to be tailored to the particular needs of identity management.

Further, it has been shown that individuals tend to leverage different identities or roles in their lives, when acting in, for example, a professional context (e.g., as an employee), in the context of a customer-company interaction (e.g., as a bank client), in the context of special interest communities (e.g., in online gaming) or in the context of their family and close friends. The structure of present technologies and the users interaction with these, however, have been found to conflict with the natural segregation of audiences and therefore blur the existence of partial identities and the privacy considerations attached to these; the natural segregation of audiences has largely been lost through present technologies such as centralised social networks and storage systems (see [vdBL10]).

Consequently, emerging technologies that strive to empower privacy-enhanced and identity management-enabled services will need to satisfy the following requirements:

- Enabling numerous privacy and identity settings for different audiences, potentially also life-long (see [Pri09]).
- Be scalable across technology platforms (e.g., across Mobile Phones, Netbooks, Tablet PCs or even Cars - via their onboard computers).
- Integrate private and professional interaction (e.g., private and professional email) and multiple partial identities (e.g., for different Mobile Services such as ticketing, banking, loyalty programs, social communities or government services) into one Mobile Device.
- Work seamlessly when private and professional services are accessed on the move, e.g., across the entertainment system of the car when driving, the private Mobile Phone, or the company Laptop.

### **22.3 The Changing Context (II): Multiple Identity Providing Stakeholders Along an Increasingly Dynamic Mobile Services Value Chain**

Looking at the ecosystem through which Mobile Service are provided, numerous stakeholders can be identified:

- There are design entities for the Central Processing Units / platforms of the devices.
- There are producers of these CPUs.
- There are handset manufacturers.
- There are Mobile Network Operators (MNOs).
- There are Service Providers, for example providing additional applications for ticketing, banking, loyalty programs, social communities or government services.
- There are Service Enablers, serving as (trusted) third parties to ensure and secure a seamless execution of Mobile Services.

Initially, the MNOs were the dominant players with regards to security and the identity involved in the provision of Mobile Services. They provided the identity of the individual user when registering with the network through the identity embedded in the SIM (Subscriber Identification Module) card. With the limited scope of identity-related Mobile Services being available in the past, security, privacy, and identity management topics were embedded into rather static environments:

- Fixed and concrete client and server components, actors and scenarios constituted the service environment.
- Single identities were sufficient for a predefined set of Mobile Services.
- Time was sufficient to develop and modify client and server components when new security, privacy and identity challenges arose.

At present, the other stakeholders along the Value Chain of Mobile Services are becoming increasingly involved: Handset manufacturers interact with their users

directly via online client portals and provide applications for them (e.g., Blackberry.net and the Blackberry App World, Apple iTunes, Nokia OVI, etc). MNOs are moving in similar directions and a plethora of new Service Providers offer their applications via the open interfaces provided to them by the handset manufacturers.

Many of these direct interactions with the end-user of the Mobile Devices include security, privacy and identity management issues.

At present, however, the industry approach to managing these issues is highly fragmented. Many stakeholders intend to provide proprietary solutions on proprietary Secure Elements and prefer to secure access to the private data of their end users for their proprietary purposes. In order to tackle this trend, any intention to provide a consistent approach for security, privacy and identity management for Mobile Devices would need to be flexible and would need to integrate with the rapidly changing context of the Mobile Services industry. The presently developing industry context can be described as follows:

- Heterogeneous scenarios are omnipresent and security requirements vary strongly.
- Multiple identities (partial identities) will need to be empowered, without the predefined set of Mobile Services being known. Hence, rule-awareness rather than fixed requirements need to be implemented in order to embed security, privacy and identity management-aware behaviour in the overall system of Mobile Services.
- The equipment needs to be context-aware and flexibly follow the overarching rules under different situations.
- Time is insufficient to develop and modify client and server components when new security, privacy and identity challenges arise. Hence, the security, privacy and identity management needs will “co-evolve with the [...] steadily changing context into which it is embedded” [Pri08a].

This need for flexibility in providing security, privacy and identity management will increase even further in the future. In addition, open interfaces will be needed for Mobile Services that leverage multiple stakeholders at the moment of delivery and higher storage capacities will be required for more data intensive services. Hence, future solutions in this area will need to be highly dynamic/adaptive to ever changing environments:

“This will consider unknown equipment, actors, and heterogeneity of space. The definition of SoS will result in security policies. The client and the server will know the policies [..., and] take a “flexible and secure,” “pervasive and secure,” “resilient and secure,” “recoverable and secure” character, depending on the situation.” [Pri08a]

Given that Smart Private Mobile Devices need to be designed according to the above described dynamic context in which they are used, the question arises how Mobile Services can provide security, privacy and identity management for the multiple stakeholders along the Value Chain, the multiple technology platforms involved in servicing the end user (e.g., the CPU of a Laptop, a Netbook, a Mobile Phone or

the navigation and entertainment system of a car) and the highly dynamic environment.

Briefly, one could ask “How can a Mobile Ecosystem for secure, privacy-enhanced and identity management-enabled services be designed and provided?”

It has been shown elsewhere that modularisation through interfaces, the standardisation of these and the collaboration across corporate boundaries are essential characteristics to provide innovative systems that comply with highly complex and dynamic market and technology environments [Ber09]. Secure Elements (SEs) are such highly modularised and standardised technologies. Different SEs can tailor for different levels of flexibility, openness and storage capacity (see [Figure 22.1](#)).

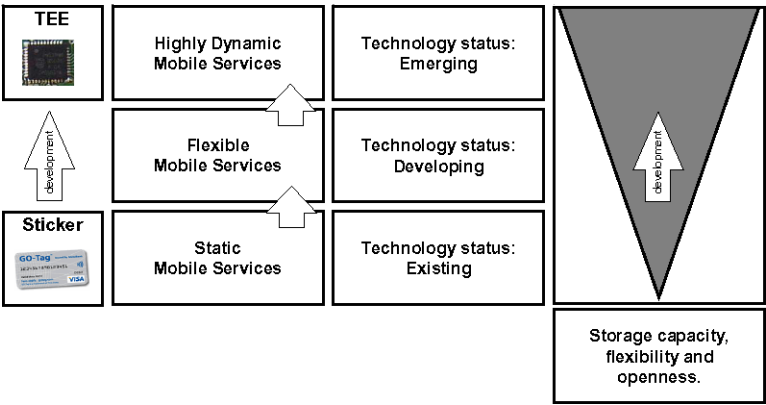


Fig. 22.1: Secure Elements between static and dynamic Mobile Services.

Because SEs play an important role for identity management and privacy in the context of Mobile Services, and because a choice between different SEs directly influences the extent to which identity management and privacy can be provided, a brief description of exemplary Secure Elements is given below.

## 22.4 Technologies for Identity Management and Privacy Enhancement: Secure Elements

Secure Elements (SEs) are platforms, particularly for Mobile Devices, on which Applications can be installed, personalised and managed. Increasingly, this can be done over-the-air (OTA). With recent<sup>1</sup> technology developments, OTA provisioning

<sup>1</sup> Recent referring to the introduction of such services from 2006/7 onwards.

of Applications is done via a Trusted Service Manager (TSM). This helps to adapt formerly static SEs more flexibly for new Mobile Services and Applications.<sup>2</sup>

Further, SEs are seen to potentially provide a “safe resort” for value-intensive, critical Applications that use significant professional and private data, especially as the environment for Mobile Devices and the services provided via these is increasingly challenged with risks of data theft [Bac10], espionage [Rt10], and security breaches [Web10].

On a conceptual level, SEs can be categorised into three different areas:

- Removable SEs (e.g., Stickers, Secure Micro SD cards and UICCs<sup>3</sup>)
- Non-removable SEs (e.g., embedded SEs)
- SEs from a combination of software programs on dedicated hardware (e.g., Trusted Execution Environments).<sup>4</sup>

The history of Secure Elements and the capabilities of Smart Cards and Tokens in the context of Secure Dynamic Mobile Services has already been analysed elsewhere (see [Pri08a]).

Further, the security in embedded systems and the different virtualisation technologies have been analysed and the usability aspects of Secure Environments have been commented upon, and the applicable cryptography has been revised (see [SSP08]).

In essence, it has been shown that SIM / UICC cards, for example, have advantages for operator-specific, static and highly secure identification tasks, and embedded security systems. In comparison, virtualisation technologies are more applicable for highly dynamic service provisioning (see [Pri08a]).

In addition to the well established Smart Card/SIM/UICC technologies, some additional Secure Element technologies have emerged successfully in the Mobile Ecosystem, whilst others have not spread so widely. As a result, the present context of SE technologies for Mobile Devices appears as shown in [Figure 22.2](#).

A selection of the above-mentioned SEs also supports increasingly Dynamic Mobile Services – without sacrificing security.

Technologies such as Secure  $\mu$ SD cards, Stickers and selected embedded Secure Elements have seen rather wide uptake in the market, because they enable new Mobile Services in the Value Chain for established stakeholders and/or were accessible to new players for the establishment of new Mobile Service concepts.<sup>5</sup>

Secure Micro SD cards ( $\mu$ SD) and the Trusted Execution Environment are of particular relevance, as they combine increased security with increased flexibility. Because these two SEs can also be used as storage and processing platforms for the

<sup>2</sup> For practical examples, see, e.g., [www.venyon.com](http://www.venyon.com) or [www.smarttrust.com](http://www.smarttrust.com)

<sup>3</sup> A UICC is a UMTS Integrated Circuit Card, i.e., a type of Subscriber Identification Module (SIM) used in 3G UMTS devices.

<sup>4</sup> In the case of the TEE, the SE consists of a physical module, e.g., a partition of the CPU and software embedded into this physical module (e.g., a secure Operating System). For a detailed elaboration on the different categories of SEs.

<sup>5</sup> Also see [Mob10] for a detailed analysis of the Mobile Value Chain – with particular attention to Mobile Financial Services.

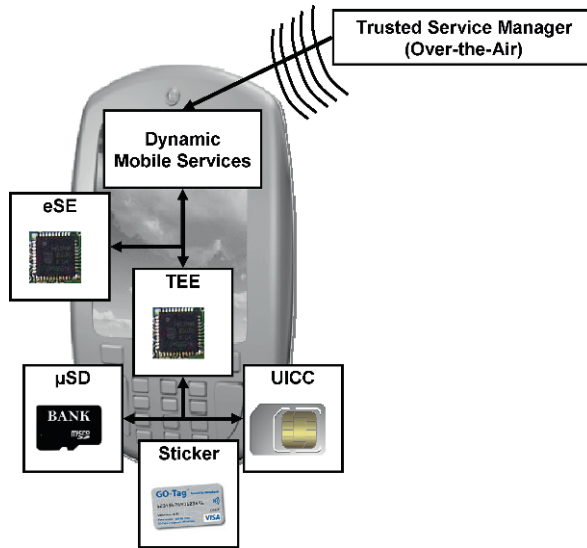


Fig. 22.2: Various Secure Elements as “Private World” in Mobile Devices.

identification of individuals and their credentials, they are particularly relevant for privacy-enhanced and identity-management-enabled services that need to be highly secure and flexible.

Further, they offer largely open interfaces within their architecture in order to promote a rapid uptake by existing and new stakeholders along the Mobile Services value chain.

However, initiatives to embed Trusted Platform Modules into Mobile Devices have not succeeded on a wide basis. Apparently, the economic incentive for the different stakeholders along the value chain of the Mobile Services industry remained unclear and fragmented business interests along the value chain were not orchestrated for a systemic solution [Mob10]. Neither has the potential option to integrate an additional Smart Card reader into Mobile Devices found wide acceptance.<sup>6</sup>

In order to review existing Secure Element technologies and subsequently introduce emerging technologies for future innovations in the field, selected technology examples are introduced in the following sections.

<sup>6</sup> The additional effort and costs for the handset manufacturer, who largely operates based on the requirements of the Mobile Network Operators (MNOs), were not consistently required and called for along the Mobile Services Value Chain. Similarly, a requirement for an additional Smart Card reader was not commonly agreed upon from all players in the Value Chain.

## **22.5 Present Secure Element Technologies: UICCs and Stickers**

### ***22.5.1 The Universal Integrated Circuit Card (UICC) and the Smart Card Web Server***

In second generation mobile Networks (2G), the SIM card was the physical Smart Card used to control access of mobile devices to the MNO network. In third generation networks (3G), this physical component is called UICC. UICCs use Java-based Operating Systems and they increasingly include additional Applications such as information-on-demand menus, SIM-based browsers, mobile banking Applications or ID credentials for other Mobile Services.

For 15 years, the SIM/UICC has been an important SE in the world of mobile communications. It is unique in taking over the global market as the one exchangeable authentication token in GSM and UMTS networks. In the past few years, this SE has been enhanced by many more functionalities than simple user authentication. The SIM/UICC has evolved to become a central medium for the storage and administration of user data in the provider network.

Moreover, the SIM/UICC has been enabled to exchange information directly with the mobile phone user and the provider network using additional mechanisms such as the Card Application Toolkit (CAT) and Over the Air (OTA) communication, and also interacts with web technologies such as HTML (hypertext markup language) pages and HTTP (hypertext transfer protocol) by leveraging the Smart Card Web Server (SCWS) technology. This enables the SIM/UICC to protect countless data such as music, video clips, purchased ring tones, and personal data on the Mobile Device. Hence, the UICC is developing to become the internet-enabled network node that seamlessly integrates into other IP networks. At present, it is the premier link that creates confidence among the mobile phone user, the network operator, and services providers in the operators network.

The specific capabilities of UICCs in the context of dynamic and secure mobile services has already been analysed elsewhere. Here, advantages of UICCs for operator-specific, static and highly secure identification tasks have been pointed out and have been compared with embedded security systems and virtualisation (see [Pri08a]). In essence, it was shown that:

“Smart Cards and Tokens provide high security in a mobile and flexible manner. Embedded Security Mechanisms and Virtualization may provide significant processing power for security relevant applications” [Pri08a]. Also, a combination of these two, still independent capabilities could be combined into “a system which can cover all levels of security, be static as well as flexible and highly performing. As an example, such systems could provide Smart Cards or Tokens for mobile devices which can store different identities and assist in using selected ones of these for different services such as payments, bookings or participation in online communities. The Embedded Security Mechanism would assist in decoding and processing the data stored

on the Smart Card or Token, thus making the overall system secure and highly performing.” [Pri08a]

With regards to security, privacy and identity management, the UICC ranks as highly secure, but it does not consciously offer technologies that drive privacy protection and enable partial identities for services outside the direct identification towards the mobile network. In fact, the most important role of the UICC remains what it was initially conceptualised for: identifying the individual user to one service partner – the Mobile Network Operator.

At present, the identity that the UICC provides is not yet intensively leveraged for other Mobile Services in order to provide their additional offerings to the end-user.

### ***22.5.2 The Sticker as Example for Static Mobile Service Identities***

“Stickers” are self-adhesive contactless cards or tags that can be stuck on the back of Mobile Devices. Although being very similar to a standard contactless Smart Card, they have a specifically designed antenna combined with a ferrite backing layer to cut distortion to and from the phones components and its radio signal. With this antenna, Stickers connect to Near Field Communication (NFC) terminals to enable NFC payments.

Currently, there are two forms of Stickers: Passive Stickers<sup>7</sup>, which are not connected to the Handsets Application execution environment, i.e., the Operating System (OS), and Active Stickers, which are connected to the OS, for example via Bluetooth.

Passive Stickers are widely available at present<sup>8</sup> and, for example, serve as empowering technologies for the payment of small amounts, just as a debit or credit card would do.<sup>9</sup> Active Stickers<sup>10</sup> that would allow more flexibility in the provisioning and adaptation of partial identities are being tested for market introduction and mass-market availability is expected in the course of 2010-11.

With regards to security, privacy and identity management, Passive Stickers provide one set of identification data (e.g., a debit/credit card number) that is inflexible and serves the need for privacy in the same manner as a normal credit card would.

---

<sup>7</sup> “Passive Stickers” have no connection to the Operating System of the mobile device. Therefore, they neither allow dynamic Application management, be it by a TSM for Application updates or by the consumer for additional services via a phones user interface, nor do they offer the full NFC use case range or the provisioning of multiple application after they are distributed.

<sup>8</sup> Passive stickers have been mass-produced in millions of units since Q1 2009 for payment and loyalty Applications.

<sup>9</sup> For practical examples, see, e.g., [www.blingnation.com](http://www.blingnation.com).

<sup>10</sup> “Active Sticker” are connected to the handset application execution environment, for example, via a Bluetooth connection. Hence, they would be able to offer flexibility regarding the flexible adoption of different identities. OTA provisioning and life cycle management by a TSM is possible for Active Stickers because of their connection to the phone. The end customer may also manage his/her MFS Applications via the phones user interface.



Active Stickers are an emerging technology that may be capable of offering flexibility with regards to the adaptation of partial identities in the future, depending on successful market trials.

## 22.6 Emerging Secure Element Technologies: Trusted Execution Environments and the Privacy Challenge

The appeal of SEs will be particularly high to the respective service providers if they rapidly, easily and seamlessly integrate with applications provided by Third Parties in the market place. Quick diffusion can be expected, if the Secure Elements in the Front-End enable these Third Parties to embed security, privacy and identity-management into their solutions ad hoc. Further, a pre-certification of the Secure Elements with regard to security, privacy and identity-management may still enhance market acceptance, as it would provide independent solution and application providers with a “dock-on” method to security, privacy and identity-management. To achieve this goal, clear and open interfaces will be essential. Further, a combination of hardware, software, interfaces and protocols needs to inter-play in order to enable the secure storage and usage of credentials for increasingly sophisticated Mobile Services. For this, technologies such as the ARM TrustZone may be leveraged, because of their dominant design in the marketplace for Mobile Device platforms.

The so-called Trusted Execution Environments (TEEs) are striving to provide the above-mentioned characteristics.

In order to remain highly flexible and adaptive to changes in the environment of Mobile Services, TEEs strive for independency from the Rich-OS. This is particularly important as increasingly open Rich-OS systems diffuse in the Mobile Devices, e.g., Googles Android. Modern TEE approaches can be used on a wide range of TrustZone systems, especially if they are equipped with a clean and easy to understand integration interface. Here, reference drivers can be leveraged that help TEEs integrate with specific Operating Systems, such as Googles Android.

TEEs provide security, privacy and identity-management solutions that enable new types of services. TEEs address the need for flexible, powerful and efficient security solutions in various forms of Mobile Devices. Among others, TEEs can, for example, be based on ARM TrustZone enabled chipsets (so called SoCs). TEEs utilise ARM TrustZones division of the SoC into two distinct areas, a “Public World” and a “Private World” as shown in [Figure 22.3](#). TEEs then provide open interfaces in order to enable the development of dedicated applications with security, privacy and identity-management capabilities.

In this concept of “Public and Private Worlds,” the TEEs encapsulate security-, privacy and identity-management-relevant parts of an application in the dedicated “Private World.” Those parts of the application that are not security-, privacy- or identity-management-relevant remain in the “Public World.”

Two clear and open interfaces between the “Public” and the “Private World” – the so called TEE Client Application Protocol Interface (API) and the TEE Internal API

- enable application providers to dock-on to the concept. Leveraging these two API empowers them to offer secure services to the market without having to go into the details of security and privacy protection or the specifics of identity-management.

The TEE Client API and the TEE Internal API follow a lightweight approach, meaning they are easy to use and easy to understand. Hence, developers can concentrate on the design of their business logics. TEEs are also integrated into different SoCs in order to diffuse quickly to the different Mobile Devices. In short, these two interfaces and the concept of TEEs offer a highly modularised, non-complex and easy to use Secure Element on Mobile Devices, which empowers rapid deployment and constant adaptation of security-, privacy- and identity-management-enhanced solutions for e.g., Mobile Phones, Netbooks, Tablet PCs or even Car navigation and entertainment systems.

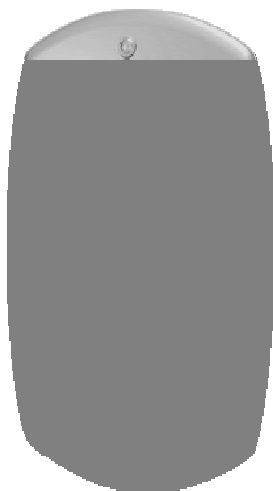


Fig. 22.3: Overview of the “Public” and “Private World” and the Interfaces in TEEs.

In order to complement the existing Secure Elements in an adequate way, TEEs are characterised by four points: High performance, low footprint, provable security and certifiability.

Looking further into modern applications on Mobile Devices and their value for the individual User (e.g., mobile banking applications, mobile social networking and mobile loyalty programs etc.), protecting these interactions and assuring the adequacy of the information exchanged via Mobile Devices becomes increasingly clear. At present, two security gaps remain for Mobile Devices, especially Mobile Handsets:

- The input of data in a trusted manner (e.g., without interference between the act of typing on the keyboard or the touchscreen), and

- the output of data in a trusted manner (e.g., without the display of manipulated data over the screen of Mobile Devices), see [Figure 22.4](#).

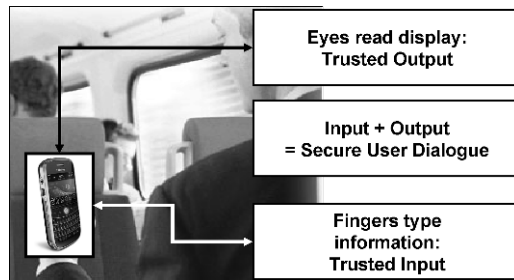


Fig. 22.4: Trusted Input + Output = Secure User Dialog.

TEEs can provide a Secure User Dialogue by assuring that any input will be transmitted in a secure way via the “Private World.” Also, TEEs can ensure that the keypad is disconnected from the “Public World” while the individual is reading trusted information on the screen. However, the usage of the secure keypad functions does not yet prevent an attacker from writing a “Public World” sniffer that could grab keypad data that is intended for the “Private World.” Therefore, a secure keypad application needs to be combined with a secret that is stored in the “Private World.” Nevertheless, such solutions assist in providing the individual with input and output in an even more secure manner than all other existing SEs at present, and also a privacy- and identity-management-enhanced one.

## 22.7 Technologies for Secure and Dynamic Mobile Services and the Privacy Challenge in Highly Dynamic Environments

Based on the above examples of Secure Element technologies, it becomes evident that static technologies such as the Passive Stickers provide a Secure Element for selected Mobile Services (e.g., NFC functionalities of Credit Card payments), but do not correspond to an environment that would call for a highly flexible provisioning of partial identities. As Passive Stickers, they have one or a set of preinstalled identities (e.g., a credit card number), but cannot be provided with new, partial identities over-the-air in a flexible manner.

Further, privacy is only partially provided, as the Passive Sticker would interact with a terminal, which would then route the communication through an additional network. Hence, there is no “private” end-to-end communication between the in-

dividual and the recipient of the message (e.g., a payment via Credit Card), but processing networks are involved in “getting the message across.”

In between the Passive Stickers and the TEE, one can position a recently introduced technology called the Secure Micro SD Card.

With secure  $\mu$ SD cards, flexibility is given and partial identities as well as privacy can be assured. Here, partial identities can be provided over-the-air for different relationships and audiences (e.g., a partial identity for traveling with frequent flyer programs, another partial identity for interacting with a bank, a third partial identity for customer loyalty programs, et cetera). Further, these partial identities can be combined with unique keys at both ends of the communication channel (i.e., VPN-like architectures), so that a communication channel that is linked to one of the partial identities remains “private” because it is encrypted and can only be read by the counterpart for this partial identity and not the processing network in between (Nota bene: Also see the PrimeLife demonstrator with the PrimeLife Application running in a Privacy-PIN protected Private World on a  $\mu$ SD Card, encrypting and decrypting messages to a specific head hunter account).

This could, for example, be used for the provisioning of “private” health information from an insurance company to a patient via mobile phones, which is impossible today in the U.S. because privacy cannot yet be assured.


	<div>Sticker</div> <div></div>	<div>TEE</div>
Highly dynamic	No	Yes
Trust: A Trusted Secure Element / Environment	Yes	Yes
Identity: A specific communication channel for the partial identity	No	Yes
Privacy: Secure communication, only for the individual	Partially	Yes
Anonymity: Unlinkability of the interaction to the individual	Possibly	Possibly

Fig. 22.5: Dynamics, Security and Privacy relating to the analysed technologies.

For TEEs, most categories are fulfilled in the same manner as for the  $\mu$ SD cards. In addition, TEEs can provide a direct link to the hardware, e.g., a mobile phone and its keypad and display, and can thus assist in even making the input and output of information trustworthy, private and identity-related (see previous section for more details).

For example, a secure User Interface (UI), which includes a secure display and keypad, will assure that the input, e.g., an amount for a money transfer or the need for a new medicine prescription, can only be read by the respective Application that is linked to the partial identity (e.g., a bank account or pharmacy), and is then encrypted and wired through the network to the recipient in a privacy-enhanced manner.

For all of these technologies, however, anonymity as an additional building block for enhanced privacy remains an open issue: For example, Mobile Devices have unique identities in the networks over which they communicate, provided by the log-on of the SIM card (i.e., the subscriber identification module) to the networks, based on the corresponding Personal Identification Number. Hence, the network can identify the individual Mobile Device. Further, the usage profile correlated to the respective device provides further insight into the individuals preferences and interests. Thus, although the above-mentioned technologies can fully or partially provide trusted, identity-related and private means of communication an interaction, the individual device, and therefore also its user, is not anonymous.

Therefore, existing and emerging Secure Element technologies already provide a valid development in the direction of secure, identity-enabled and privacy-enhanced Mobile Services.

However, certain privacy challenges remain. These should be the future directions of innovation in the area of Smart Private Mobile Devices and Services.

## **22.8 Contributions of the PrimeLife Project for the Advancement of Technologies in the Field**

The PrimeLife project advanced the technology for identity management and privacy on mobile devices in the following areas:

Firstly, it conceptualised how infrastructures for Mobile Services of the future need to be structured in order to allow for highly dynamic Mobile Services (see [Pri08a]). Here, it was found that the existing technologies do a great job for rather static Mobile Services (also see section on UICC and Stickers above) but solutions for more flexible environments are missing.

Subsequently, G&D developed the front-end for a PrimeLife demonstrator and used a secure  $\mu$ SD card as a more flexible and open Secure Element for this [Pri08b].

Applying the scenario of the eCV, the demonstrator created the following, highly dynamic Mobile Service as an interaction between the Mobile Device, a “Private World” embedded into the secure  $\mu$ SD card, and the server back-end [Pri08b, pp.24-25]:

- “A portal (i.e., a back-end server) manages potential privacy and identity management related conflicts and sends requests that contain conflicts to the “Private World” of a users Mobile Device.”
- “The user then decides whether he/she wishes to use the requesting service or not, based on logging in to her/his “Private World” (via the Privacy-PIN) in the Secure Element (e.g., Secure  $\mu$ SD Card or TEE) on the Mobile Device.”
- “The “Private World” on the Mobile Device can “freeze” the Back-end if the end user/job applicant does not accept a policy mismatch.”
- “The “Private World” on the Mobile Device can deactivate the data set on the Back-end for a selected time period, e.g., if the job applicant does not want his private data to be visible for others online because he does not want to be approached by any job offering entity.”
- “The “Private World” on the Mobile Device can directly interact with the Back-end in a secure manner (e.g., via a Virtual Private Network or via encrypted communication) for data control in the future.”
- “The “Private World” on the Mobile Device holds the essential service keys for numerous privacy- and identity-management enhanced services and is therefore the privacy and identity-controlling device in the palm of each end consumer.”
- “The “Private World” on the Mobile Device provides a secure compartment/the TEE in which customisable services are empowered and in which additional data can be stored, e.g., additional certificates to enhance the eCV even more in selected cases.”
- “The “Private World” on the Mobile Device can “glue” other Secure Elements such as the SIM, the SD card and others together, if these are needed as sources of partial identities to provide more complete identity sets for particular services.”

The PrimeLife demonstrator therefore enabled more flexible Mobile Services, distributed between the SE on the front-end Mobile Device and the back-end Server.

Nevertheless, a highly dynamic composition of Mobile Services was only partially possible and additional features of privacy (e.g., a secure User Interface, as shown in [Figure 22.4](#)) remained an open issue. Further, the secure  $\mu$ SD card as selected SE was not available in all mobile devices and was rather a plug-in solution.

Hence, G&D focused on the next level of SEs that would enable not only flexible, but highly dynamic services: Trusted Execution Environments [Pri08b]. Here, G&D provided and assured the standardisation of the TEE Client API, which enables the exchange of data between the rich OS Applications of Mobile Devices and the security- and privacy-enhanced part (i.e., the “Private World”) of the Application. Based on the standardisation in the Global Platform consortium and an open TEE Client API, developers can now define which data shall be interchanged between the “Public” and the “Private World,” and in which manner. They can do so for the Mobile Services of any stakeholder along the value chain and can also provide their applications to the Secure Element in a highly flexible manner by using over-the-air service providers that “load” the identity-management-enabled and privacy protected applications into the Mobile Devices in the field.

Through the design of this interface and its standardisation in alignment with the PrimeLife project, G&D empowered the Mobile Services environment with access



Fig. 22.6: Dynamic Mobile Services in the PrimeLife Demonstrator.

to a highly modularised, simple and easy-to-use Secure Element, which empowers rapid deployment and constant adaptation of security-, privacy- and identity-management-enhanced solutions for e.g., Mobile Phones, Netbooks, Tablet PCs or even cars.

## 22.9 The Privacy Challenge in Mobile Services and Future Directions for Innovation

The above-elaborated status quo of the Mobile Services ecosystem and the Secure Element technologies has drawn an up-to-date picture of the technical capabilities and has shown which issues in the area of security, identity and privacy are presently being solved, amongst others via the PrimeLife project.

However, selected areas remain unsolved. They can be summarised in the following roadmap for further innovation in the field:

- The requirement for highly dynamic adaptation to the ever-changing market and technology environments can be tailored for through the different SE alternatives. The less dynamic the SEs need to be, the more one can expect that existing so-

lutions will be leveraged. The more dynamic Mobile Services shall be designed, the more likely emerging SEs such as Secure  $\mu$ SD cards and TEEs will be used.

- The Security / trust requirement is inherently solved by the usage of SE technologies.
- The requirement for numerous, partial identities is predominantly addressed by the emerging SEs, such as Secure  $\mu$ SD cards and TEEs. It can be expected that Mobile Devices will leverage multiple SEs in the future, each hosting different partial identities.
- Privacy, as in the secure communication between predefined communication partners that relate to the respective partial identities (e.g., secure one-to-one communication between the end user and the eCV portal, between the end user and a banking/payment entity, or between the end user and a selected loyalty program provider such as an airline), can be assured through the emerging SE technologies.
- Anonymity, as in the unlinkability of the end user to its respective actions, remains an open issue. Here, technologies such as IDMix and Direct Anonymous Attestation can prove very valuable if they were combined with the above-mentioned SEs. These could counteract the present unique identity of Mobile Devices (e.g., via their Subscriber Identity) and the linkability of this identity to the usage profile in Applications that still largely reside in the “Public” instead of the “Private” world.
- Further, TEEs are not yet as strictly isolated as, for example, UICC/SIM cards and their certification for the different use cases still needs to be provided. In addition to the open question of anonymity, this needs to be addressed through future research.

In essence, the existing and emerging technologies are a significant step into the direction of providing more (partial) identity-related and privacy-empowered Mobile Services. Nevertheless, open points such as anonymity, certification and isolation remain to be solved in future directions of innovation.



	<div>           Sticker            </div>	<div>           TEE            </div>	
Highly dynamic	No	Yes	Different SEs expected to co-exist, as existing and future applications will vary strongly.
Trust: A Trusted Secure Element / Environment	Yes	Yes	The SE concept inherently solves the trust issue.
Identity: A specific communication channel for the partial identity	No	Yes	Partial identities either pre installed ad hoc, dynamic.
Privacy: Secure communication, only for the individual	Partially	Yes	New SEs expected to be preferred if privacy shall be stronger than in existing solutions.
Anonymity: Unlinkability of the interaction to the individual	Possibly	Possibly	Next step beyond security, privacy and identity: Anonymity.

Fig. 22.7: Remaining privacy challenges and future directions for innovation.

## Chapter 23

# Privacy by Sustainable Identity Management Enablers

Sascha Koschinat, Gökhan Bal, Christian Weber, and Kai Rannenberg

**Abstract** Telcos face an elementary change in their traditional business model. The reasons for this are manifold: Tougher regulations, new technology (most notably VoIP and open spectrum), matured core business markets (voice and messaging), new market entrants or advancing customer demands and expectations. A potential direction of this change is business models that concentrate on the exploitation and monetisation of the huge amount of customer data that results from the usage of traditional communication services (data, voice). Based on these data, telcos' long-standing relationships to their customers, and infrastructural assets and capabilities, telcos are a reasonable candidate for assuming the role of identity management service providers (IdMSPs). This chapter describes a method to evaluate privacy-enhancing IdM Services from the perspective of a telco acting as prospective IdM Service Provider. The basis for the evaluation method is formed by the concept of *Identity Management Enablers*, which are used to analyse and describe the services and scenarios on which the decision supporting method is based on.

## 23.1 Introduction

Telcos face an elementary change in their traditional business model. The reasons for this are manifold: Tougher regulations, new technology (most notably VoIP and open spectrum), matured core business markets (voice and messaging), new market entrants or advancing customer demands and expectations.<sup>1</sup> Telcos are forced into decision-making about new business models. A potential direction of this change is business models that concentrate on the exploitation and monetisation of the huge amount of customer data that results from the usage of traditional communication services (data, voice). One out of several potential future business models will be the provision of identity management services to third-party service providers. Based

---

<sup>1</sup> <http://www.Telco2.net/manifesto/>

on these data, Telcos' longstanding relationships to their customers, and infrastructural assets and capabilities, Telcos are a reasonable candidate for assuming the role of identity management service providers (IdMSPs). For these reasons, we are focussing on Telcos in the role of IdMSPs instead of other potential Internet-based service providers (e.g., Facebook, Google, Amazon). But like other organisations, Telcos have concerns about the economic motivations to invest in privacy-enhancing identity management services [FaRi08]. This chapter describes a method for the construction and application of a decision support approach that can be applied to support the decision-making process of Telcos in order to decide on investing into the provision of privacy-enhancing identity management services. The method has seven steps. Some of them are structured following established economic methods. The ongoing work described in the following sections is an initial design of this method.

Section 23.2 introduces the IdM enabler concept and provides a step by step description of the method, where each step is followed by an illustrative use case example. Section 23.3 describes further use case examples for the method. Section 23.4 gives an overview of related approaches in this area. Finally, Section 23.5 briefly discusses the benefits and limitations of the method and gives an outlook on further potential developments.

## 23.2 Economic Valuation Approach for Telco-Based Identity Management Enablers

This chapter introduces a decision support approach that can be applied by Telcos in order to decide whether to invest in the provision of privacy-enhancing identity management services. The basis for the evaluation method described below is formed by the concept of *Identity Management Enablers*. Identity Management Enablers are used to analyse and describe the services and scenarios on which the decision supporting method is based. They consist of a valuable combination of IdM related customer data assets<sup>2</sup> and functional capabilities. Data assets in this context are attributes of a user identity (e.g., end customers) such as name, place of birth, account details, and so forth. Functional capabilities are functions that process these data assets to provide IdM services (e.g., age verification, authentication). A combination of IdM related functional capabilities and identity related data assets is further called an *IdM Enabler* and should be seen as a driver for a specific IdM Service (see Figure 23.1).

The evaluation approach can be used as a decision support instrument for potential providers of privacy-enhancing IdM services and consulting agencies acting in this domain. It is focused on decision situations, where an IdM service provider (IdMSP) has to decide

- between investing in a privacy-enhancing IdM Service or not,

---

<sup>2</sup> Data collected due to the provision of traditional communication services.

- investing in which one of at least two alternative privacy-enhancing IdM services.

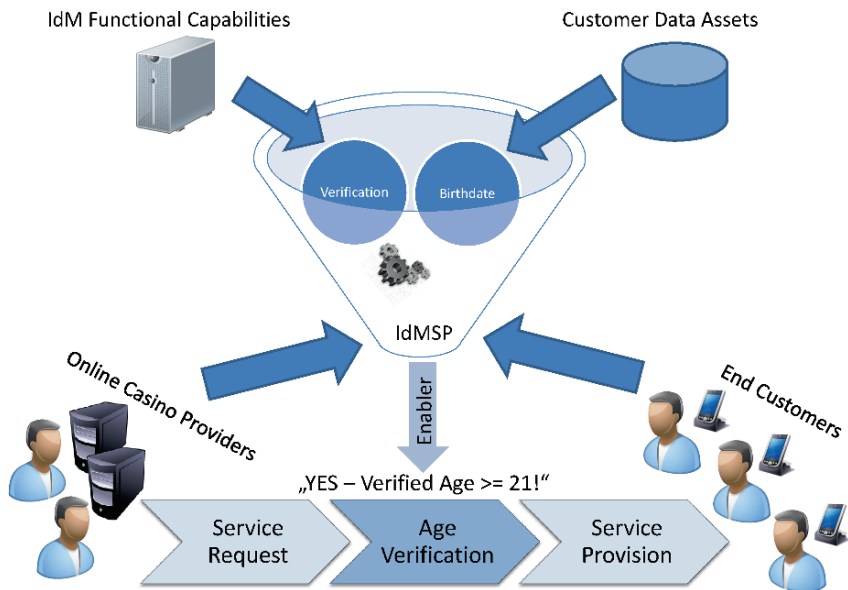


Fig. 23.1: The IdM Enabler Concept.

The method consists of the following seven process steps. These will be described in more detail in the subsequent sections:

1. Description of the baseline option and feasible delta options by scenarios.
2. Identification of each stakeholder's costs and benefits.
3. Selection of each stakeholder's key costs and benefits.
4. Mapping of each stakeholder's key cost and benefits to the IdMSP by cause-effect chains.
5. Clustering IdMSP's costs and benefits.
6. Assessment and aggregation of IdMSP's clustered costs and benefits.
7. Visualisation of IdMSP's aggregated cost and benefits.

To demonstrate each step of the approach in a more pragmatic and less abstract way, they will be consistently applied to the following exemplary use case: An IdMSP has to decide whether to invest in the provision of a new privacy-enhancing age verification service for end customers of online casino providers.

### 23.2.1 Description of the Baseline Option and Feasible Delta Options

In the first step, the IdMSP describes the status quo of the examined identity management service. This mainly comprises a description of how a specific identity management service is currently implemented in practice by other service providers. This status quo scenario is called the *Baseline Scenario (BS)*. Thus, the *Baseline Option (BO)* represents the alternative not to provide any of the available IdM services at all, and needs to be considered as one possible decision.

After the description of the Baseline Scenario, the IdMSP needs to describe all alternative implementation scenarios of the IdM service that shall be considered to enhance the Baseline Scenario. These alternative scenarios are here called the *Delta Scenarios (DS)*. Further, these Delta Scenarios must be mutually exclusive. Analogous to the Baseline Option, a *Delta Option (DO)* represents the alternative to invest in one of the Delta Scenarios.

#### Use Case - Age Verification Scenario

In this example, the decision maker (the IdMSP) identified two alternative designs for an enhanced age verification service: Delta Scenario 1 (DS 1) and Delta Scenario 2 (DS 2). In this case, the IdMSP has the following options to act in the identity management ecosystem: Invest in DO 1, in DO 2 or do not invest (BO) at all. The IdMSP's decision for the BO would leave the state of the resulting environment unchanged as shown in Figure 23.2.

In this example, the end customer of an online casino needs to provide the online casino provider with a valid proof of his age. Here, the end customer provides this information by, e.g., entering his date of birth into a special web form. This process has to be replicated for any age-based service the end customer wants to use.

Opting for DO 1 would result in the modified market situation represented by Delta Scenario 1 (DS 1) (Figure 23.3). To use the age verification service, the end customer needs to create an account with the IdMSP and needs to provide a valid proof for his date of birth. This usually will involve an external age verification process. After being successfully registered with the IdMSP, a verified legal age certificate will be provided by the IdMSP. The end customer can use this certificate at any point in time and without the involvement and the knowledge of the IdMSP in order to verify its legal age to the online casino provider. The end customer can request additional verified legal age certificates to be presented to other age-based services. Alternatively, the IdMSP could issue generic, service provider independent credentials, e.g., in the form of an anonymous credential [CL01].

Opting for DO 2 would result in a modified market situation represented by Delta Option 2 (Figure 23.4).

In Delta Scenario 2 (DS 2), the end customer needs to create an account with the IdMSP and provide a valid proof for his date of birth with the help of an ex-

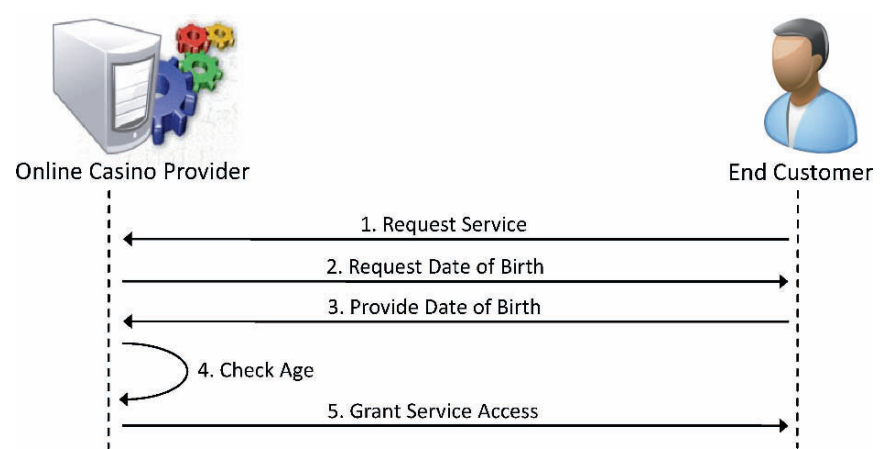


Fig. 23.2: Age Verification Baseline Scenario.

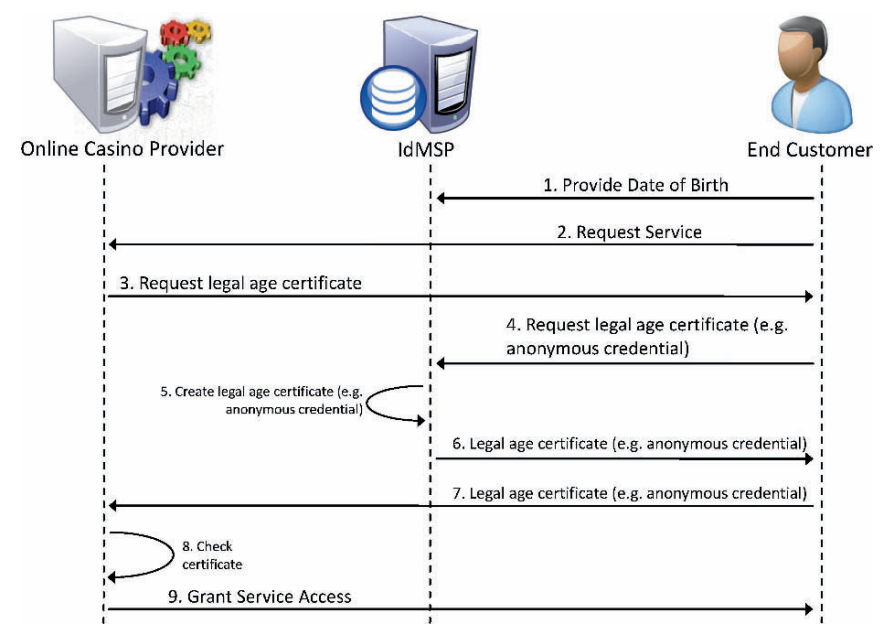


Fig. 23.3: Age Verification Delta Scenario 1.

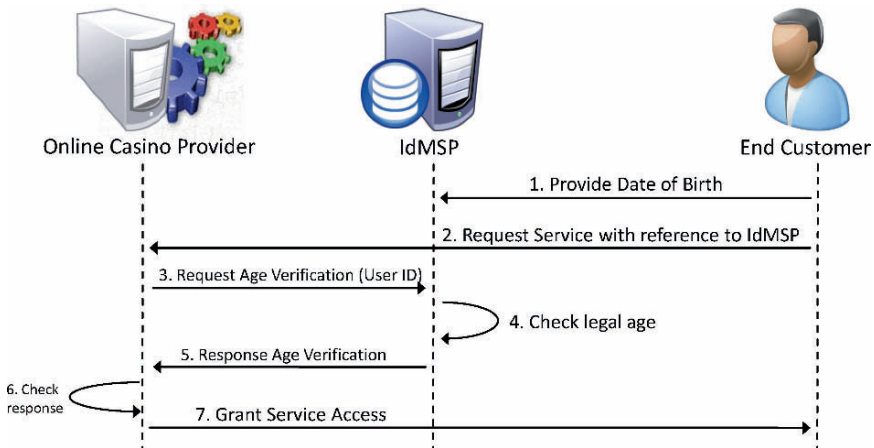


Fig. 23.4: Age Verification Delta Scenario 2.

ternal age verification process. After successful registration, when the end customer wants to use the online casino service, he provides the online casino provider a reference to his age verification provider. The online casino provider then requests the IdMSP for verified age information. Thus, the end customer is not involved to the age verification process.

### 23.2.2 Identification of each Stakeholder's Costs and Benefits Based on Delta Scenarios in Comparison to the Baseline Scenario

The anticipated impacts and the expected costs and benefits of a specific scenario are crucial factors for decision making. Therefore, the corresponding costs and benefits need to be identified for all delta scenarios. During this step, the Baseline Scenario has to be taken as the reference value (the *baseline*). That allows for the prediction and evaluation of the consequences of the Delta Scenarios in the form of costs and benefits. This step of the method can be performed by experts or by the usage of an appropriate explanatory model. As the costs and benefits of the IdMSP partially depend on the costs and benefits of the other market players (service providers, end customers), these have also to be anticipated and evaluated in this step (Figure 23.5 and 23.6).

Delta Option 1 vs. Baseline Option					
End Customer		Service Provider		IdM Service Provider	
Costs	Benefits	Costs	Benefits	Costs	Benefits
<ul style="list-style-type: none"> <li>• Additional efforts for Hardware and/or Software</li> <li>• Additional registration fees and/or charges for service usage</li> <li>• Additional efforts for IdMSP registration</li> <li>• Higher duration of transactions</li> <li>• Additional risk of data misuse by IdMSP</li> <li>• Additional risk of missing availability of a service due to failure of End Customer or IdMSP infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>• More Privacy, because of additional Data Minimisation</li> <li>• Lower risk of data misuse by Service Providers</li> <li>• Higher trust in Service Providers</li> <li>• Additional guaranteed compliance with regulation</li> </ul>	<ul style="list-style-type: none"> <li>• Fewer possibilities for commercialisation of End Customer data</li> <li>• Less information about End Customers</li> <li>• Less potential for advertising, personalisation, profiling, targeting etc.</li> <li>• Additional risk of missing availability of a service due to failure of End Customer or IdMSP infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>• Fewer efforts for infrastructure implementation and operation</li> <li>• Higher convenience for being compliant with regulations</li> <li>• Higher End Customer loyalty</li> <li>• More new End Customers</li> <li>• More revenues</li> <li>• Lower risk of payment losses through minors</li> <li>• Additional guaranteed compliance with regulation</li> <li>• Fewer efforts for End Customer support</li> </ul>	<ul style="list-style-type: none"> <li>• Additional efforts for development and operation of hardware and/or software for End Customers</li> <li>• Additional efforts for development and operation of the Service Infrastructure</li> <li>• Additional efforts for development and operation of the Business Model</li> <li>• Additional efforts for correction of incorrect age verifications</li> <li>• Additional efforts for End Customer support</li> </ul>	<ul style="list-style-type: none"> <li>• Additional Value Added Service for End Customers</li> <li>• Higher End Customer loyalty</li> <li>• More new End Customers</li> <li>• More revenues</li> <li>• Higher market entry barriers for possible business rivals</li> </ul>

Fig. 23.5: Cost-Benefit list for Delta Option 1 vs. Baseline Option.

Delta Option 2 vs. Baseline Option					
End Customer		Service Provider		IdM Service Provider	
Costs	Benefits	Costs	Benefits	Costs	Benefits
<ul style="list-style-type: none"> <li>• Less Privacy, because of additional knowledge of IdMSP about End Customers' Service Providers</li> <li>• Less Privacy, because of less control about personal data provision</li> <li>• Additional efforts for IdMSP registration</li> <li>• Higher duration of transactions</li> <li>• Additional risk of data misuse by IdMSP</li> <li>• Additional risk of missing availability of a service due to failure of Service Provider or IdMSP infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>• Higher convenience for being compliant with regulations</li> <li>• More Privacy, because of additional Data Minimisation</li> <li>• Lower risk of data misuse by Service Providers</li> <li>• Higher trust in Service Providers</li> <li>• Additional guaranteed compliance with regulations</li> </ul>	<ul style="list-style-type: none"> <li>• Additional costs for implementation and operation of interface infrastructure to IdMSP</li> <li>• Additional registration fees and/or charges for service usage</li> <li>• Additional efforts to provide incentives for End Customers</li> <li>• Fewer possibilities for commercialisation of End Customer data</li> <li>• Less information about End Customers</li> <li>• Less potential for advertising, personalisation, profiling, targeting etc.</li> <li>• Additional risk of missing availability of a service due to failure of Service Provider or IdMSP infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>• Additional business relationship to IdMSP</li> <li>• Higher End Customer loyalty</li> <li>• More new End Customers</li> <li>• More revenues</li> <li>• Lower risk of payment losses through minors</li> <li>• Additional guaranteed compliance with regulations</li> <li>• Fewer efforts for End Customer support</li> </ul>	<ul style="list-style-type: none"> <li>• Additional efforts for Service Provider support</li> <li>• Additional efforts for development and operation of the Service Infrastructure</li> <li>• Additional efforts for development and operation of the Business Model</li> <li>• Additional efforts for correction of incorrect age verifications</li> <li>• Additional efforts for End Customer Support</li> </ul>	<ul style="list-style-type: none"> <li>• Additional business relationships to Service Providers</li> <li>• Additional possibilities for new Marketing &amp; Sales-channels</li> <li>• Additional Value Added Service for End Customers and Service Providers</li> <li>• Higher End Customer loyalty</li> <li>• Higher Service Provider loyalty</li> <li>• More new End Customers</li> <li>• More new Service Providers</li> <li>• More revenues</li> <li>• Higher market entry barriers for possible business rivals</li> </ul>

Fig. 23.6: Cost-Benefit list for Delta Option 2 vs. Baseline Option.



Delta Option 1 vs. Baseline Option					
End Customer		Service Provider		IdM Service Provider	
Costs	Benefits	Costs	Benefits	Costs	Benefits
<ul style="list-style-type: none"><li>• Additional efforts for Hardware and/or Software</li><li>• Additional registration fees and/or charges for service usage</li><li>• Additional efforts of IdMSP registration</li><li>• Higher duration of transactions</li><li>• Additional risk of data misuse by IdMSP</li><li>• Additional risk of missing availability of a service due to failure of End Customer or IdMSP infrastructure</li></ul>	<ul style="list-style-type: none"><li>• More Privacy, because of additional Data Minimisation</li><li>• Lower risk of data misuse by Service Providers</li><li>• Higher trust in Service Providers</li><li>• Additional guaranteed compliance with regulations</li></ul>	<ul style="list-style-type: none"><li>• Fewer possibilities for commercialisation of End Customer data</li><li>• Less information about End Customers</li><li>• Less potential for advertising, personalisation, profiling, targeting etc.</li><li>• Additional risk of missing availability of a service due to failure of End Customer or IdMSP infrastructure</li></ul>	<ul style="list-style-type: none"><li>• Fewer efforts for infrastructure implementation and operation</li><li>• Higher convenience for being compliant with regulations</li><li>• Higher End Customer loyalty</li><li>• More new End Customers</li><li>• More revenues</li><li>• Lower risk of payment losses through minors</li><li>• Additional guaranteed compliance with regulations</li><li>• Fewer efforts for End Customer support</li></ul>	<ul style="list-style-type: none"><li>• Additional efforts for development and operation of hardware and/or software for End Customers</li><li>• Additional efforts for development and operation of the Service Infrastructure</li><li>• Additional efforts for development and operation of the Business Model</li><li>• Additional efforts for correction of incorrect age verifications</li><li>• Additional efforts for End Customer support</li></ul>	<ul style="list-style-type: none"><li>• Additional Value Added Service for End Customers</li><li>• Higher End Customer loyalty</li><li>• More new End Customers</li><li>• More revenues</li><li>• Higher market entry barriers for possible business rivals</li></ul>

Fig. 23.7: Key Costs and Benefits for Delta Option 1 vs. Baseline Option.

Delta Option 2 vs. Baseline Option					
End Customer		Service Provider		IdM Service Provider	
Costs	Benefits	Costs	Benefits	Costs	Benefits
<ul style="list-style-type: none"><li>• Less Privacy, because of additional knowledge of IdMSP about End Customers' Service Providers</li><li>• Less Privacy, because of less control about personal data provision</li><li>• Additional efforts for IdMSP registration</li><li>• Higher duration of transactions</li><li>• Additional risk of data misuse by IdMSP</li><li>• Additional risk of missing availability of a service due to failure of Service Provider or IdMSP infrastructure</li></ul>	<ul style="list-style-type: none"><li>• Higher convenience for being compliant with regulation</li><li>• More Privacy, because of additional Data Minimisation</li><li>• Lower risk of data misuse by Service Providers</li><li>• Higher trust in Service Providers</li><li>• Additional guaranteed compliance with regulations</li></ul>	<ul style="list-style-type: none"><li>• Additional costs for implementation and operation of interface Infrastructure to IdMSP</li><li>• Additional registration fees and/or charges for service usage</li><li>• Additional efforts to provide incentives for End Customers</li><li>• Fewer possibilities for commercialisation of End Customer data</li><li>• Less information about End Customers</li><li>• Less potential for advertising, personalisation, profiling, targeting etc.</li><li>• Additional risk of missing availability of a service due to failure of Service Provider or IdMSP infrastructure</li></ul>	<ul style="list-style-type: none"><li>• Additional business relationship to IdMSP</li><li>• Higher End Customer loyalty</li><li>• More new End Customers</li><li>• More revenues</li><li>• Lower risk of payment losses through minors</li><li>• Additional guaranteed compliance with regulations</li><li>• Fewer efforts for End Customer support</li></ul>	<ul style="list-style-type: none"><li>• Additional efforts for Service Provider support</li><li>• Additional efforts for development and operation of the Service Infrastructure</li><li>• Additional efforts for development and operation of the Business Model</li><li>• Additional efforts for correction of incorrect age verifications</li><li>• Additional efforts for End Customer Support</li></ul>	<ul style="list-style-type: none"><li>• Additional business relationships to Service Providers</li><li>• Additional possibility for new Marketing &amp; Sales-channels</li><li>• Additional Value Added Service for End Customers and Service Providers</li><li>• Higher End Customer loyalty</li><li>• Higher Service Provider loyalty</li><li>• More new End Customers</li><li>• More new Service Providers</li><li>• More revenues</li><li>• Higher market entry barriers for possible business rivals</li></ul>

Fig. 23.8: Key Costs and Benefits for Delta Option 2 vs. Baseline Option.

### **Use Case - Age Verification Scenario**

The identified costs and benefits are described in a way that they express the expected economic changes that result from introducing a Delta Scenario. Each identified cost and benefit has an influence on the overall benefit that is expected.

#### ***23.2.3 Selection of Key Costs and Benefits for each Stakeholder***

To reduce the overall complexity, in this step, the IdMSP has to reduce the set of costs and benefits to a subset of key costs and key benefits for each stakeholder. The IdMSP excludes all costs and benefits he does not consider relevant for its decision making.<sup>3</sup> As the following steps of the method (steps 4 - 7) are based on this reduced subset of costs and benefits, the selection of key costs and key benefits is crucial for the overall result.

### **Use Case - Age Verification Scenario**

Figure 23.7 and Figure 23.8 show the result of the exemplary application of this step of the method. Costs and benefits that are considered as less relevant are crossed out.

#### ***23.2.4 Mapping of each Stakeholder's Key Cost and Benefits on IdM Service Provider by Cause-Effect Chains***

The key costs and benefits identified for each stakeholder in a Delta Scenario have to be mapped to the IdMSP by cause-effect chains. The central idea of cause-effect chains is to create a model of the resulting cost and benefits, which particularly considers their interdependencies. A single cost or benefit of a market player causes economic effects on the respective market player itself and on all other players of that ecosystem. The aim of the cause-effect chains is to let all economic effects of the other market players flow into the IdMSP's costs and benefits. As a result, the IdMSP will get a set of mapped costs and benefits representing the economic consequences caused by the other market players.

---

<sup>3</sup> Note that the result of this step is highly dependent on the decision maker's individual valuation of each cost and benefit.

Use Case - Age Verification Scenario

All key costs and benefits derived in Step 3 (Section 23.2.3) will now be mapped step by step to the IdMSP (Figure 23.9 and Figure 23.10):

- Mapping end customer’s key costs and key benefits to other costs and benefits of the end customer.
- Mapping end customer’s costs and benefits to costs and benefits of the service provider.
- Mapping end customer’s and service provider’s costs and benefits to costs and benefits of IdMSP.

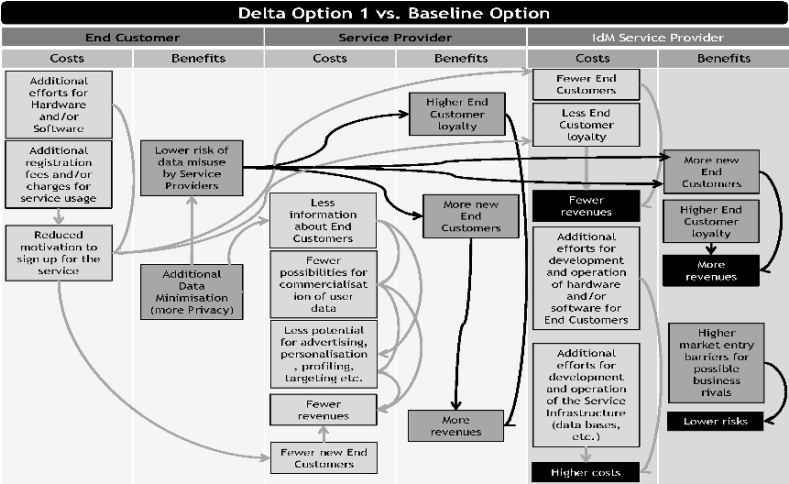


Fig. 23.9: Cause-Effect-Chain for Delta Option 1 vs. Baseline Option.

The last two columns in the tables in Figure 23.9 and Figure 23.10 show the results of this step.

23.2.5 Clustering of Mapped IdM Service Provider Costs and Benefits

After mapping all costs and benefits to the IdMSP, Step 4 will usually result in a large set of different costs and benefits with a variety of scale units. To reduce complexity and ease the process, clustering by equal scale units or dimensions such as revenues, costs, or risks, to a (as small as possible) set of decision-relevant factors, is needed.

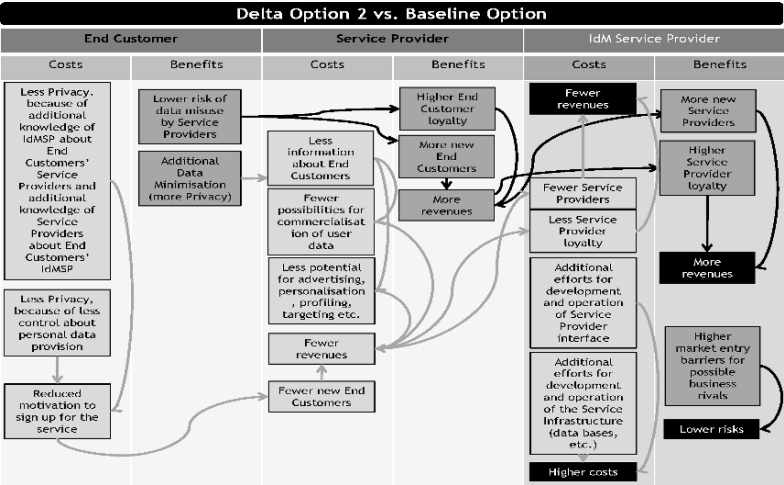


Fig. 23.10: Cause-Effect-Chain for Delta Option 2 vs. Baseline Option.

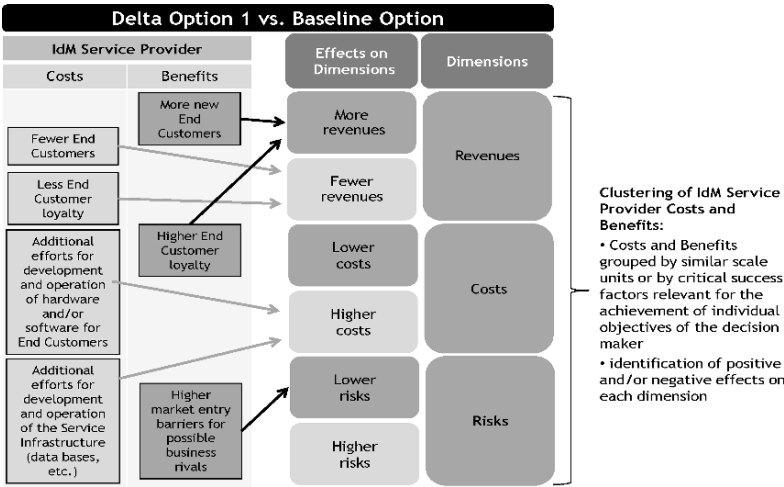


Fig. 23.11: Clustering Costs and Benefits of Delta Option 1 vs. Baseline Option.

Use Case - Age Verification Scenario

The clustering of all costs and benefits by similar scale units or by critical success factors that are relevant for the achievement of IdMSPs individual objectives, results in a set of costs and benefits that is easier to handle. With this clustering, the effects of a group of similar costs or benefits will be represented by a single effect in an aggregated form (see Figure 23.11 and Figure 23.12). For example, more new

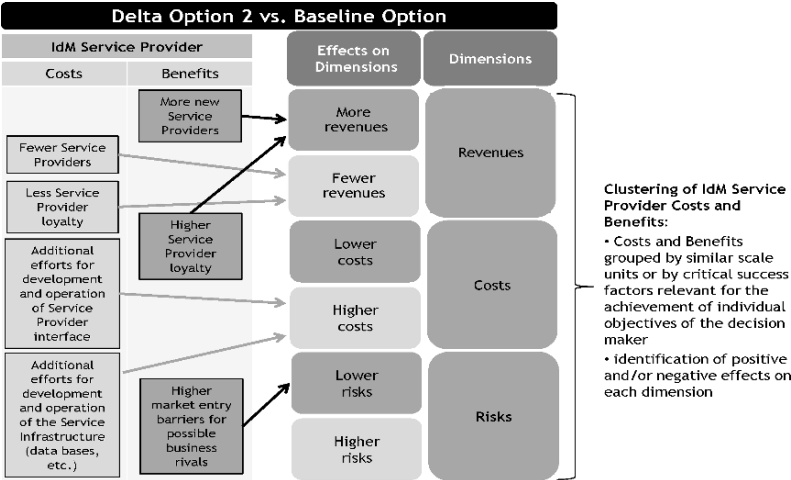


Fig. 23.12: Clustering Costs and Benefits of Delta Option 2 vs. Baseline Option.

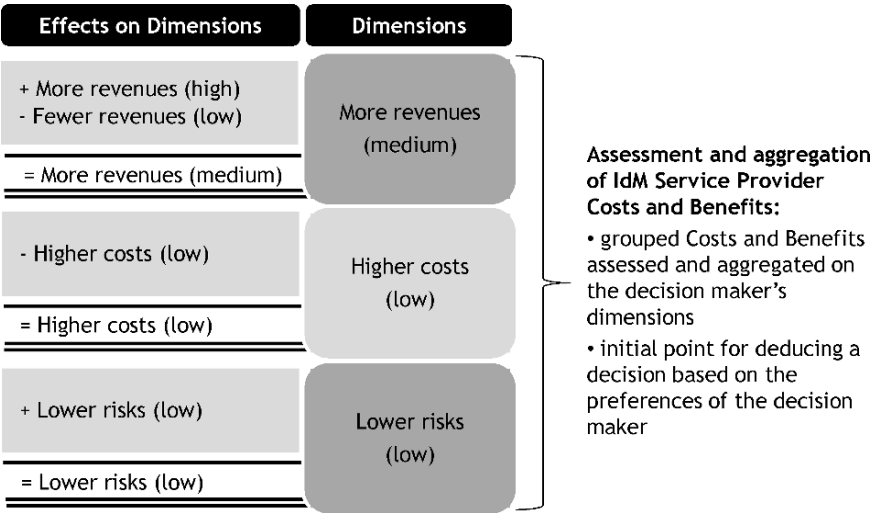


Fig. 23.13: Aggregating Costs and Benefits of Delta Option 1 vs. Baseline Option.

service providers and a higher degree of service provider loyalty will result in more revenue.

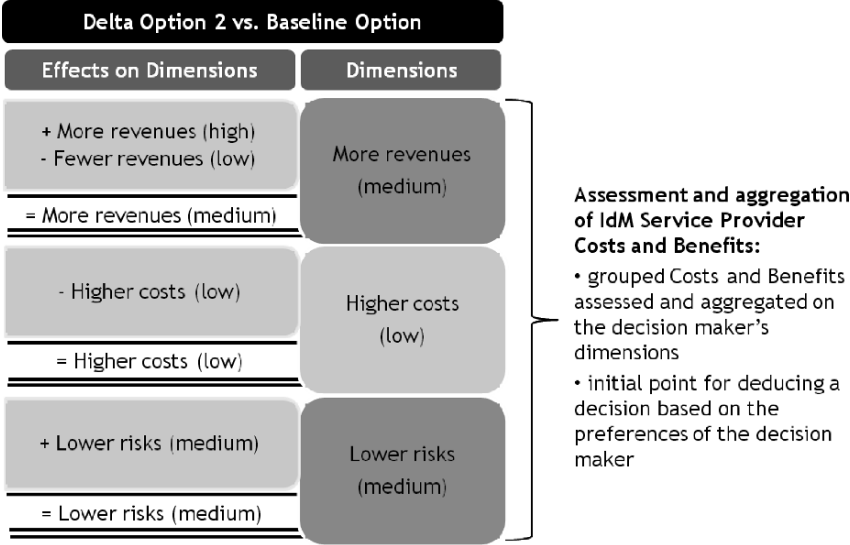


Fig. 23.14: Aggregating Costs and Benefits of Delta Option 2 vs. Baseline Option.

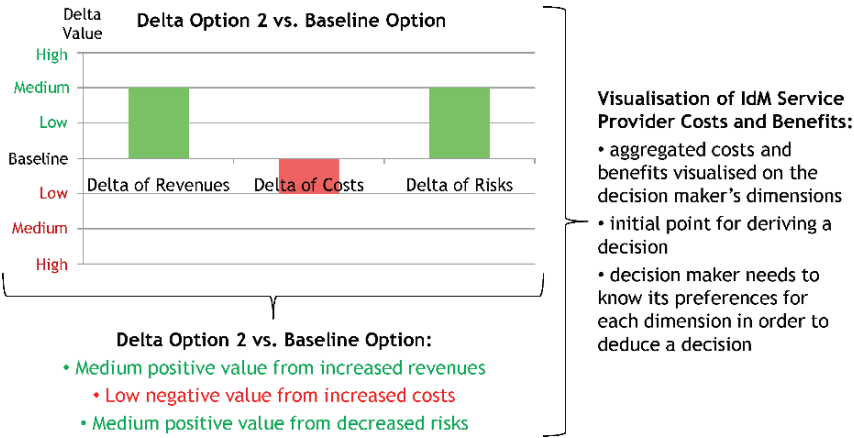


Fig. 23.15: Visualising Costs and Benefits of Delta Option 1 vs. Baseline Option.

23.2.6 Assessment and Aggregation of Clustered IdM Service Provider costs and Benefits

The effects resulting from the cost and benefit clustering in Step 5 (Section 23.2.5) can be positive or negative and of different importance to the IdMSP. Therefore the effects need to be aggregated to an overall effect for each of the chosen dimensions.

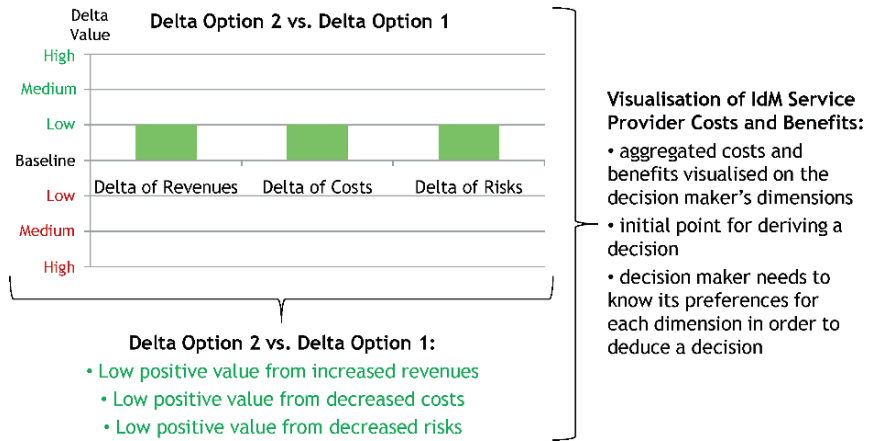


Fig. 23.16: Visualising Costs and Benefits of Delta Option 2 vs. Baseline Option.

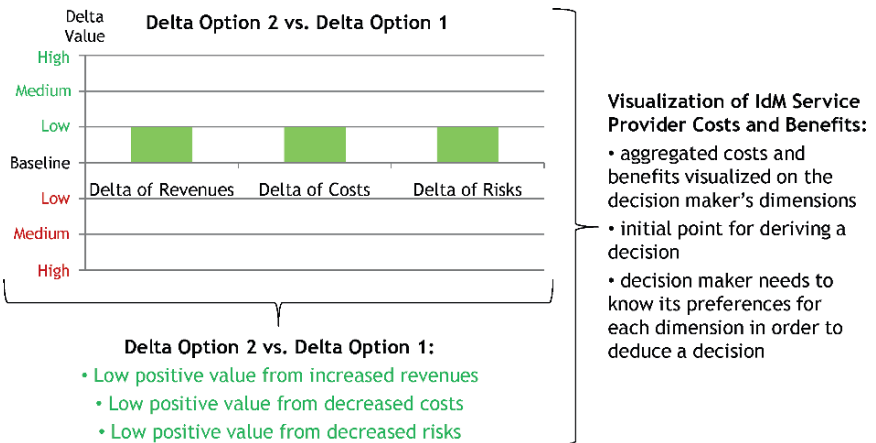


Fig. 23.17: Delta Option 2 vs. Delta Option 1.

During the aggregation, each effect needs to be individually weighted by the IdMSP. Where applicable, this can be done by adding concrete values or ranges of values for each effect, but usually the aggregation will be based on appropriate scales or grades defined by experts of the IdMSP, such as very good (+ +), good (+), medium (0), bad (-), and very bad (- -).

### Use Case - Age Verification Scenario

Based on the results of Step 4 and Step 5, the IdMSP now assesses the intensity of each dimension influencing effect by using the abstract value classes high negative (- - -), medium negative (- -), low negative (-), Baseline (0), low positive (+), medium positive (+ +), and high positive (+ + +). For example, for the comparison between the DS 1 and the BS (Figure 23.13), the IdMSP rates the effect *more revenues* as *medium positive* and the effect *fewer revenues* as *low negative*. In the end, the IdMSP expects the effect *more revenues* to have *low positive* intensity, when it opts for DO 1. Based on the IdMSP's preferences for each dimension and the results shown in Figure 23.13, the IdMSP can now deduce the decision whether or not it should provide its age verification service as represented by DO 1.

### 23.2.7 Visualisation of Aggregated IdM Service Provider Costs and Benefits

Finally, the aggregated costs and benefits will be visualised in order to further simplify complex decision situations to support the IdMSP.

### Use Case - Age Verification Scenario

Visualisation example (see Figure 23.15 and Figure 23.16): Based on the results of Step 6 (23.2.6), the IdMSP should also value the relative advantages of its DOs as shown in Figure 23.17.

## 23.3 Description of the Identity Management Scenarios

The method presented in Section 23.2 can be applied to a variety of identity management services. Step 1 of the method requires the IdMSP to describe the Baseline Scenario and the Delta Scenarios. In Section 23.1, a first identity management enabler scenario with its baseline and delta scenarios have been presented (age verification). To give more examples for the design of Baseline and Delta Scenarios, this section presents two additional identity management enabler scenarios (authentication, privacy policy enforcement) that can be evaluated with the method. For both scenarios, the Baseline Scenario and two Delta Scenarios are presented.



### 23.3.1 Authentication

Authentication is an essential identity management function in online and offline scenarios. Due to its nature, implementations of authentication mechanisms have to be reliable and secure. Therefore, different authentication types exist for different scenarios. In the following, three possible authentication designs are presented, the baseline option and two possible delta options.

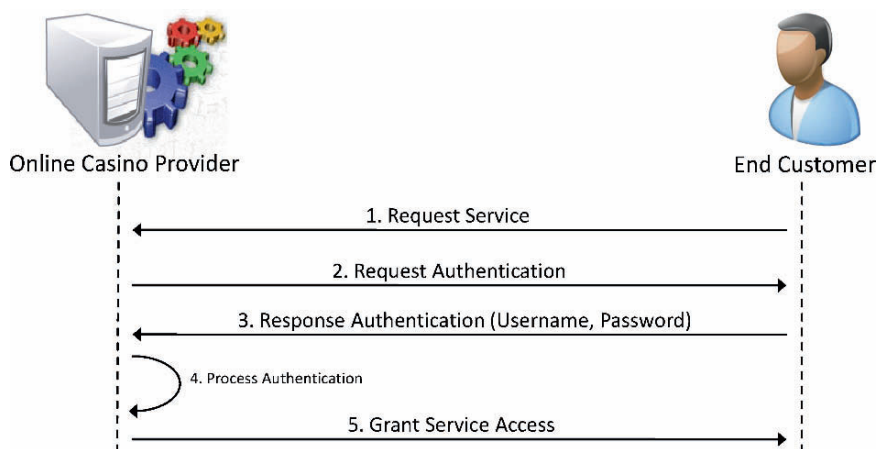


Fig. 23.18: Authentication Baseline Option.

#### 23.3.1.1 Baseline Option - End Customer Provides Username & Password

The baseline option illustrated in Figure 23.18 represents the most commonly used authentication scheme in online scenarios. Before a session starts, the end customer provides his username (pseudonym) together with the password to the service provider. The service provider then authenticates the user. Though, the service is enabled by the user providing the authentication credentials (IdM data asset) and the service provider processing the authentication (IdM functional capability).

### **23.3.1.2 Delta Option 1 - Telco Forwards Authentication Code to End Customer**

A more sophisticated authentication scheme is Delta Option 1, illustrated in Figure 23.19. The most known implementations of this multi-factor authentication scheme can be found in online banking scenarios (mobile TAN, smsTAN, mTAN). In Delta Option 1, the Telco is the trusted third-party IdMSP. In the first step, after the end customer requests the service, the service provider requests the Telco to forward an authentication code to the mobile phone of the end customer (1). The authentication code is generated randomly on-the-fly and is valid for a short time frame. After the end customer receives this authentication code (2), he presents this second authentication credential to the service provider (the first authentication step was providing a username and password combination to the SP).

This authentication scheme requires that the service provider and the Telco negotiate a commonly used identifier for the respective user in the initial registration phase. Here, the essential IdM data asset (mobile phone number of the end customer) comes from the Telco, the IdM functional capability is on the service provider's side (processing the authentication). This scheme follows *Privacy by Design* principles, since the phone number of the end customer is not shared with the service provider.

### **23.3.1.3 Delta Option 2 - Telco Provides Authentication Data**

Delta Option 2 (Figure 23.20) is a generalised single sign-on scenario. The user has an (SSO-)account with the Telco. If he wants to consume a specific service, he authenticates to the Telco. The Telco then provides the user an authentication token, which the end customer then forwards to the service provider. In this scheme, the essential IdM functional capability (processing the authentication) is implemented and provided by the Telco.

## **23.3.2 Privacy Policy Enforcement**

Privacy policies are an essential instrument for the end customers to express their privacy preferences. Policy enforcement mechanisms ensure that these policies are followed. This section presents three different types of policy enforcement implementations.

### **23.3.2.1 Baseline Option - Manual Policy Enforcement by the End Customer**

Figure 23.21 simply illustrates the most common approach for a user to control the flow of personal data. There is no actual configuration of privacy policies and no

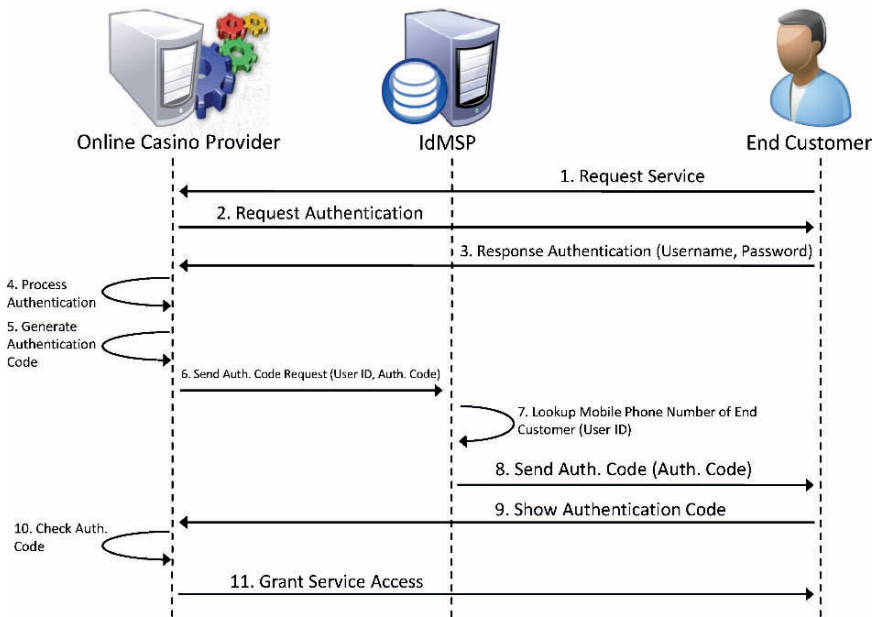


Fig. 23.19: Authentication Delta Option 1.

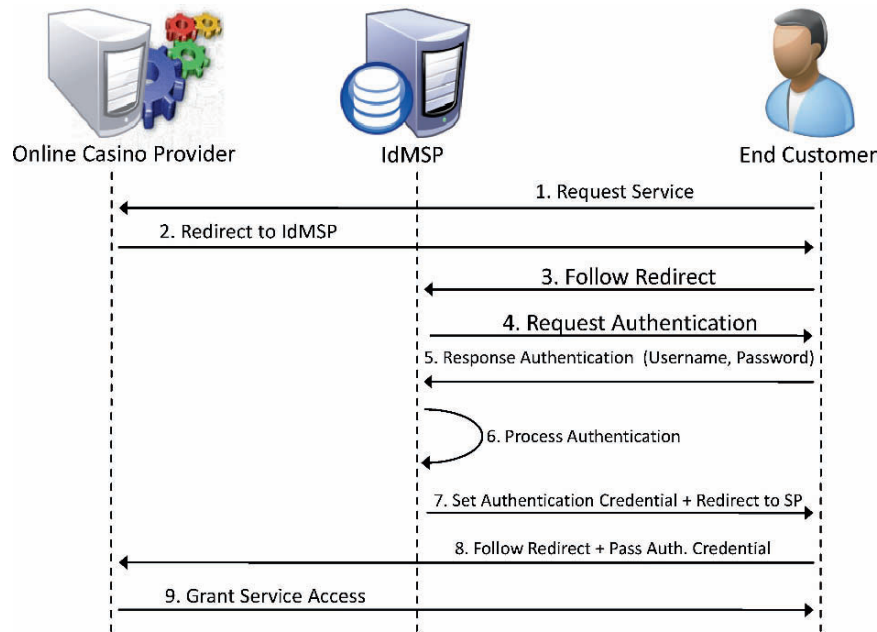


Fig. 23.20: Authentication Delta Option 2.

dedicated process for enforcing policies. The end customer selectively provides the data that he wants to share with the service provider. Obviously, this approach is not very flexible and scalable, but still the state of the art.

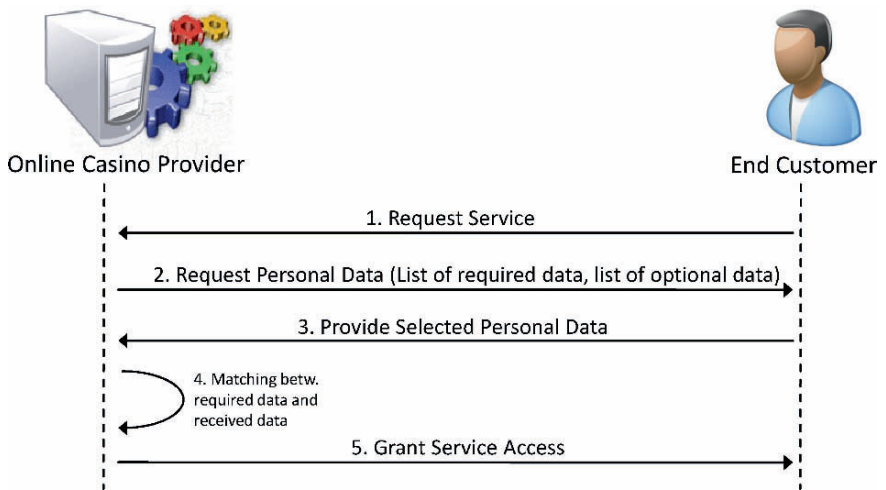


Fig. 23.21: Privacy Policy Enforcement Baseline Option.

23.3.2.2 Delta Option 1 - Service Provider Enforces Privacy Policy

In Delta Option 1 (Figure 23.22), the user provides his personal data together with a privacy policy (data handling policy) to the service provider. The SP handles the provided data as specified in the privacy policy. This variant is applicable to scenarios where the service provider is a potential provider of identity data to third parties.

23.3.2.3 Delta Option 2 - Policy Enforcement by the Telco

Figure 23.23 illustrates a policy enforcement design derived from PrimeLife results of different work packages. There is a dedicated Policy engine at the Telco’s side where the end customer can create and configure (or upload) his individual privacy policy (1). When a service provider requests personal data from the end customer, the policy enforcement point (PEP) checks this request against the user’s privacy

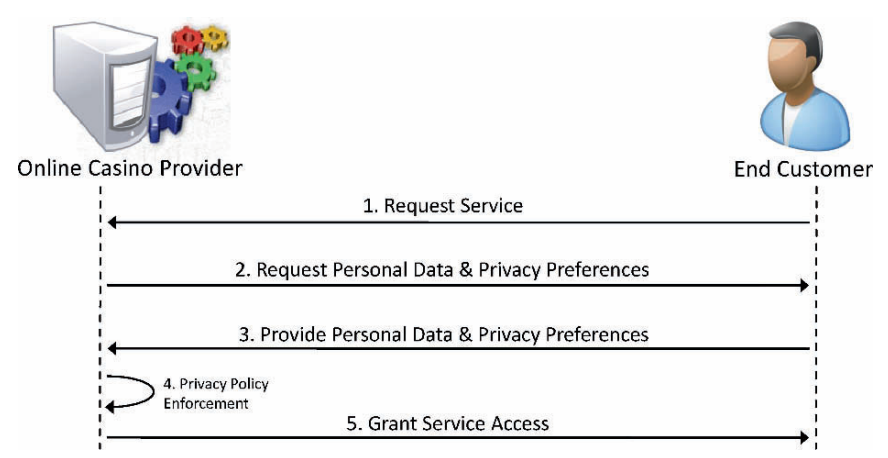


Fig. 23.22: Privacy Policy Enforcement Delta Option 1.

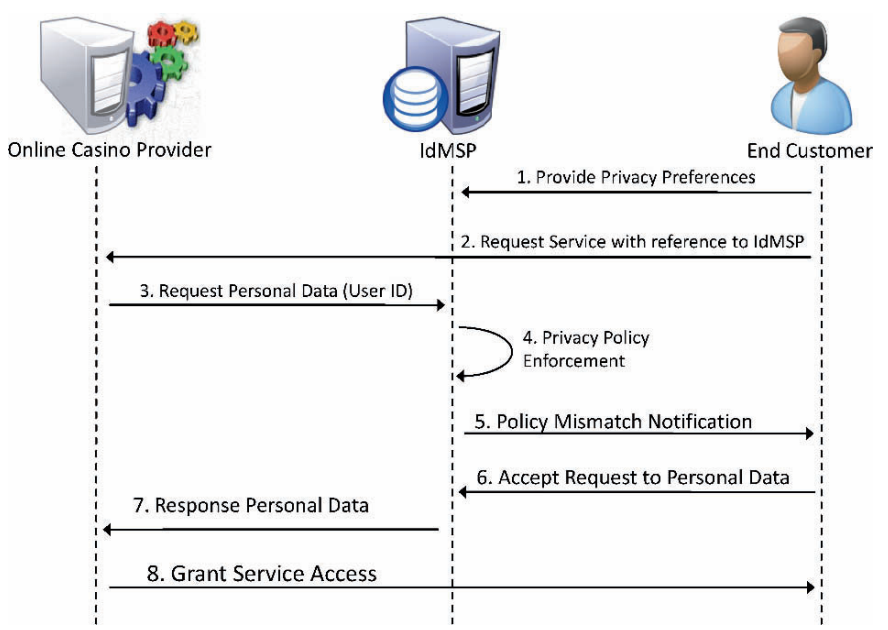


Fig. 23.23: Privacy Policy Enforcement Delta Option 2.

policy. In case of a mismatch, the end customer is informed about this (e.g., push notification to his mobile phone, 2). He then can decide whether to provide the data anyhow or to insist on his policy configuration (3). In the first case, the policy filtered data is provided to the service provider (4). In this option, both the IdM data assets (personal data) and the IdM functional capability is provided by the Telco.

## 23.4 Related Work

There is a considerable body of related work on economic issues of privacy-enhancing identity management by the project PRIME (Privacy and Identity Management for Europe) [PRI]. This work demonstrates both the potential of privacy-enhancing identity management technology as a tool for ensuring and enforcing customers' privacy in everyday transactions and the means by which it can be applied by enterprises to their everyday business.

Addressing the enterprise perspective, Fairchild and Ribbers [FR11] take up some existing concerns about the economic motivations for organisations to invest in privacy and identity management in general and explore a few intrinsic technology adoption drivers for enterprises. They propose the use of a cost benefit analysis by enterprises in order to decide whether to invest in the implementation of privacy and identity management technologies under consideration of expected changes such as costs, risks, trust, image and revenues. These are some examples of business-related effects that should be considered when evaluating or deciding to invest in privacy-enhancing identity management.

Zibuschka, Rannenberg, and Kölsch [ZRK11] explored privacy and identity management in a specific application domain: They provided a set of economic and regulatory requirements for the commercialisation of privacy-enhanced location-based services that could also be adapted as requirements for the commercialisation of more general privacy-enhancing identity management services. Commonly, in such service scenarios, different parties, e.g. a mobile network operator, an application provider and an end customer, need to interact with each other. Each stakeholder has different interests and requirements, assets and capabilities, as well as constraints and limitations. These stakeholder-depending factors influence the utility gainable by the other stakeholders in providing or consuming a service. Hence, the involved stakeholders, their utility-influencing factors and their interdependencies should also be considered when evaluating or deciding to invest in privacy-enhancing identity management services.

Kölsch, Zibuschka, and Rannenberg [KZR11] derived features from a wide range of application prototype scenarios that a privacy-friendly identity management system should support in order to fulfill the stakeholders' interests and requirements. Depending on the feature-supporting characteristics of a system, each stakeholder gains a different utility from providing or consuming it. Thus, also the different feature characteristics should be considered when evaluating or deciding to invest in a privacy-enhancing identity management system.

We extend this line of related work by proposing a structured method for evaluating privacy-enhancing identity management services.

## 23.5 Summary and Future Work

We have developed a method to evaluate privacy-enhancing IdM Services from the perspective of a Telco acting as prospective IdM Service Provider. Some of the seven steps of the method are structured following established economic methods. The major goal of our approach is to develop a simple method with a good trade-off between quality of the method's results and the effort needed in carrying out the work. To test our method, we compared several IdM service scenarios employing our valuation method and will do some more in the future. The tests showed the need for a very detailed and precise description of the scenarios. Making the scenarios more detailed and precise of course also helps to improve their understanding.

To further develop the method the following work is planned:

- Intensive testing of the method on real world use-cases.
- Further examination of the economic viability of privacy-enhancing Telco based IdM Services.
- Enhancement of the method based on the basic model of normative decision theory.
- Enhancement and improvement of each step by more sophisticated methods and concepts and for more intensive focus on privacy-related effects.
- Simplification of the applicability by predefined and selectable components for each step of the approach (e.g. predefined and selectable costs and benefits, cause-effect chain elements).
- Reducing possible errors caused by subjectivity of the decision maker.

## References Part V

- [APS02] Paul Ashley, Calvin Powers, and Matthias Schunter. From privacy promises to privacy management: a new approach for enforcing privacy throughout an enterprise. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, pages 43–50, New York, NY, USA, 2002. ACM.
- [Bac10] C. Bachfeld, D.; Mulliner. Risiko smartphone: Spionageangriffe und abzocke auf android und iphone. *c't - Magazin fuer Computertechnik*, 20:80–82, 2010.
- [BDS04] Michael Backes, Markus Duermuth, and Rainer Steinwandt. An algebra for composing enterprise privacy policies. In *Proceedings of 9th European Symposium on Research in Computer Security (ESORICS)*, volume 3193 of *Lecture Notes in Computer Science*, pages 33–52. Springer, September 2004.
- [Ber09] Marc-Michael Bergfeld. *Global Innovation Leadership: The strategic development of worldwide innovation competence*. BOD, Norderstedt, 2009.
- [BHTB05] James Backhouse, Carol Hsu, Jimmy C. Tseng, and John Baptista. A question of trust. *Commun. ACM*, 48:87–91, September 2005.
- [BNP09] Laurent Bussard, Anna Nano, and Ulrich Pinsdorf. Delegation of access rights in multi-domain service compositions. *Identity in the Information Society*, 2(2):137–154, December 2009.
- [BT10] Thomas de Buhr and Stefan Tweraser. My Time is PRIME Time. pages 69–91, 2010.
- [BZW06] Sruthi Bandhakavi, Charles C. Zhang, and Marianne Winslett. Super-sticky and de-classifiable release policies for flexible information dissemination control. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 51–58, New York, NY, USA, 2006. ACM.
- [CDK05] George Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed Systems. Concepts and Design*. Addison Wesley, 4 edition, 2005.
- [CK05] Luis Felipe Cabrera and Chris Kurt. *Web Services Architecture and Its Specifications: Essentials for Understanding WS-\**. Microsoft Press, 2005.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.
- [CLS11] Jan Camenisch, Ronald Leenes, and Dieter Sommer, editors. *PRIME – Privacy and Identity Management for Europe*, volume 6545 of *Lecture Notes in Computer Science*. Springer Berlin, 2011.
- [FR11] Alea Fairchild and Piet Ribbers. *Privacy-Enhancing Identity Management in Business*, chapter 7, pages 107–129. Volume 6545 of Camenisch et al. [CLS11], 2011.
- [KR00] David P. Kormann and Aviel D. Rubin. Risks of the passport single signon protocol. *Comput. Netw.*, 33:51–58, June 2000.



- [KZR11] Tobias Kölsch, Jan Zibuschka, and Kai Rannenber. *Privacy and Identity Management Requirements: An Application Prototype Perspective*, chapter 28, pages 723–744. Volume 6545 of Camenisch et al. [CLS11], 2011.
- [Mob10] MobeyForum. Alternatives for banks to offer secure mobile payments, whitepaper of the mobeyforum, 2010.
- [MS09] Sebastian Meissner and Jan Schallaböck. Requirements for privacy-enhancing service-oriented architectures. Public project deliverable H6.3.1, PrimeLife Consortium, November 2009.
- [PRI] PRIME. Privacy and Identity Management for Europe. <https://www.PRIME-project.eu/>.
- [Pri08a] PrimeLife WP6.2. Infrastructure for trusted content. In M.-M. H. Bergfeld, W. Hinz, and S. Spitz, editors, *PrimeLife Deliverable D6.2.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2008.
- [Pri08b] PrimeLife WP6.3. Advancement and integration of concepts for secure and dynamic creation of mobile services. In M.-M. H. Bergfeld and S. Spitz, editors, *PrimeLife Deliverable D6.3.1*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, 2008.
- [Pri09] PrimeLife WP1.3. Requirements and concepts for identity management throughout life. In Katalin Storf, Marit Hansen, and Maren Raguse, editors, *PrimeLife Heartbeat H1.3.5*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, November 2009.
- [Rt10] C. Ritten. Ausgespült: Sicherheit von apps für android und iphone. *c't - Magazin fuer Computertechnik*, 20:86–91, 2010.
- [SSP08] Joachim Swoboda, Stephan Spitz, and Michael Pramteftakis. *Kryptographie und IT-Sicherheit*. Vieweg+Teubner, 2008.
- [vdBL10] Bibi van den Berg and Ronald Leenes. Audience segregation in social network sites. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *SocialCom/PASSAT*, pages 1111–1116. IEEE Computer Society, 2010.
- [W3C02] W3C. A P3P preference exchange language 1.0 (APPEL1.0), 2002.
- [W3C06] W3C. The platform for privacy preferences 1.1 (P3P1.1) specification, 2006.
- [Web10] V. Weber. Zwickmühle: Der Streit um Blackberry-Sicherheit. *c't – Magazin für Computertechnik*, 20:144–161, 2010.
- [ZRK11] Jan Zibuschka, Kai Rannenber, and Tobias Kölsch. *Location-Based Services*, chapter 25, pages 665–681. Volume 6545 of Camenisch et al. [CLS11], 2011.

## **Part VI**

# **Privacy Live**

## Introduction

One of PrimeLife's main objectives is to make "privacy live" by raising awareness concerning privacy and security risks as well as to present solutions to maintain one's private sphere. The project contributes to this ambitious aim by transferring the mature results of the project into practice. There are various means to achieve this, and PrimeLife has applied a mixture of different approaches. For instance, PrimeLife has devoted itself to support education in the field of privacy and identity management: Together with the International Federation for Information Processing (IFIP), we conducted two summer schools for students and experts to exchange information on research questions and possible ways to tackle them. Also, this book represents one of the channels PrimeLife has chosen to convey its messages. Further, it turned out that PrimeLife became a major player in the cooperation with several other European and national projects working on privacy and identity management. All projects involved could profit a lot by the synergies stemming from this cooperation.

This chapter focuses on three important areas in which PrimeLife contributed to success in improving privacy and identity management: PrimeLife's Open Source contributions, the projects contributions to the standardisation discussion, and finally best practice solutions that emerged from PrimeLife's work as well as the cooperation with so many interested people during the project's lifetime, as explained in the following.

There are a number of different reasons why people and companies are releasing Open Source software. These include the fundamental belief that software should by principle be freely available, a means to tap into a whole community of free developers, because one would like to offer one's work to other users, or as a publication much like research papers are published. For software that implements security and privacy mechanisms, an important reason is certainly that publishing the source of a software allows for the review of an implementation so that users can be ensured that there are no hidden trapdoors and that the implemented algorithms are really providing security. In addition, people are often not willing to spend money for security (software) even if they see a potential need for it. SSH and SSL are examples of security protocols where implementations are freely available, are now widely adopted and, in fact, have enabled the growth of commerce over the Internet!

The example of SSH and SSL exhibits another important reason for Open Source: to drive adoption of infrastructure components and standards. Indeed, here Open Source activities and standardisation go hand in hand. Both help to remove technical barriers, open up new markets, and enable new economic models. At the same time, standards and free implementations of them increase opportunities for product differentiation and competition and services.

PrimeLife has made available a number of Open Source components to share its results and to allow them to be used by other parties. Initially, PrimeLife wanted to interact with some of the existing Open Source activities to have them include PrimeLife's results in their projects. This quickly turned out to be unfeasible resource-wise. Instead, the PrimeLife project decided to share its results as stand-

alone pieces and instead invest its resources in making these pieces as usable as possible. The project hopes that other projects will pick them up, experiment with the concepts, and hopefully incorporate them into real applications. Some of the code is experimental while other code is quite stable and (limited) support is offered by the authors. In Chapter 24 we give an overview of PrimeLife's Open Source contributions.

As a second way to make privacy live, PrimeLife has contributed to standardisation organisations. The main focus here was on the relevant ISO working groups and the W3C-related bodies. Within ISO, PrimeLife concentrated on JTC 1/SC 27 working group 5, in particular on the Framework for Identity Management and the Privacy Reference Architecture. As W3C was a partner in PrimeLife, the project took advantage of the W3C working style and thus organised a number of workshops for interacting with the internet development community. Besides this, we have also engaged with other standardisation bodies such as some of the OASIS committees or the Internet Engineering Task Force (IETF). The standardisation work that the PrimeLife project has already done and considers that still needs to be done is described in Chapter 25.

During PrimeLife's work on privacy and identity management, it became apparent that the current state-of-the-art information technologies used for privacy-relevant data processing is certainly not satisfactory: In many cases, it neither matches the provisions of European data protection regulation nor does it address society's and individuals' needs for maintaining privacy throughout a full lifetime. Quite the contrary – society, with today's IT systems, seems to be ill-prepared for the challenges ahead of us; instead of protecting the people's privacy, we will have to face additional risks. PrimeLife was approached by several other projects and discussed its vision of privacy for life with many stakeholders. Part of its work consisted of elaborating requirements and recommendations for all stakeholder groups involved in the complex area of privacy and identity management. In Chapter 26, we present a selection of best practice solutions that address different stakeholders. They tie together a few ideas provided as Open Source components, some material offered to standardisation initiatives, and various recommendations from PrimeLife deliverables and discussions with other projects. All of these approaches belong to PrimeLife's legacy, which can and will survive even after the end of the project.



## Chapter 24

# Open Source Contributions

Jan Camenisch, Benjamin Kellermann, Stefan Köpsell, Stefano Paraboschi, Franz-Stefan Preiss, Stefanie Pötzsch, Dave Raggett, Pierangela Samarati, and Karel Wouters

### 24.1 Introduction

Privacy protection tools can be characterised by the number of parties that have to cooperate so that the tools work and achieve the desired effect [Pfi01]: Some privacy protection tools can be used stand-alone, without the need for the cooperation of other parties. Others require that the communication partners use the same tools. Some tools only function when being supported by an appropriate infrastructure that quite often is currently not in place.

For all of these categories, one finds Open Source tools. Tools that one can just use include numerous browser extensions, such as CookieSafe, CSLite, NoScript, Ghostery, and also file encryption software such as TrueCrypt. Tools allowing the user to communicate anonymously also can be employed after being installed on the user's computer, but need the cooperation of other users or servers on the Internet, such as AN.ON, TOR, I2P, and GnuNET. Next, among the tools that require the cooperation of the communication partners is encryption software such as GnuPG, OpenPGP, and OpenSSL. Finally, there are a couple of Open Source identity management frameworks available (Higgins, OpenID) as well as access control components (OpenSAML and some XACML implementations) which offer a (limited) form of privacy if used properly.

PrimeLife has produced a survey of the Open Source landscape, which is available from <http://www.primelife.eu/> (Deliverables D3.4.1 and D3.4.2). In this chapter, we describe the software tools that the PrimeLife project has made available on <http://www.primelife.eu/results/opensource/>. For these, we also describe what related tools are available.

The software tools that PrimeLife has produced range from research prototypes to almost product-quality components. A few such as Identity Mixer (Idemix) have their origins in the earlier PRIME project<sup>1</sup>, but most are implementations of concepts that were conceived in PrimeLife and can be seen as the project's practical

---

<sup>1</sup> <http://www.prime-project.eu/>

research results. We present here a selection of PrimeLife's Open Source contribution.

## 24.2 Social Software

Social software such as web forums, social networks (e.g., Facebook, LinkedIn and wikis) have become very popular. In all these media, users share lots of personal information with the provider and with the other users of the media and often even with the entire internet community. Indeed, it has become hard if not impossible for users to know and control who has access to their shared data. PrimeLife has developed and published a number of different solutions to alleviate this.

We have developed our own social network site Clique<sup>2</sup> that allows users to easily determine who is the audience of their posts, i.e., to set the *access control policy* the provider must enforce for their post. This assumes of course that the provider is trusted to enforce the policy and not to leak their data. If one does not have this trust, one could use PrimeLife's *Scramble!* browser plug-in. This plug-in encrypts all submitted data before it gets posted to the provider in such a way that only one's friends can read the data. Thus, the access control policy is enforced with encryption.

Considering online forums, we have developed a privacy awareness tool (*Personal Data MOD*) that informs users what data they are about to disclose to whom. Furthermore, we have developed a privacy-enhancing access control systems for forums. It is based on access control policies and anonymous credentials and empowers the forum user to specify who can access her thread or forum post. We have implemented both of these prototypes for the phpBB forum engine.

In the following we describe these tools in more detail.

### 24.2.1 Clique – Privacy-Enhanced Social Network Platform

Clique is a modification of the Elgg social networking platform (cf. Section 2.2.3). Clique provides users with a social network platform that enables them to keep control over their privacy. This includes, for example, fine grained access control and configuration of multiple faces (e.g., family, personal, professional) that can be used for interactions with other users. When posting a data item, e.g., name, birthday or profile photo on the site, the user can define for every single other user whether they should be able to see it or not.

Clique achieves this with the following features:

---

<sup>2</sup> <http://clique.primelife.eu/>

- Collections – contacts are organised in collections, roughly corresponding with social circles. Users can form different instances by defining close friends, family, colleagues, former school-friends, etc.
- Flexible access control to content – all content contains attribute certificate policies based on moving collections and contacts in a plain and easy to use graphical user interface.
- Visual audience indicators – all content is labelled by icons showing who has access to the information.
- Fading relations – depending on the activity of one's contacts, these users slowly disappear. At first this happens through visual indicators (coloured border around user icon), later by closing access to one's data from the automatically de-friended contact.

### ***24.2.2 Scramble! – Audience Segregation by Encryption***

Existing social network providers, such as Facebook and MySpace, implement access control for users' data. These mechanisms offer no protection against the providers themselves, as these have access to all users' information. To address this, we have developed a model [BKW09] and implemented a Firefox extension named Scramble!<sup>3</sup> that allows not only for the definition of access control rules for audience segregation, but also for the enforcement of users' access control preferences by using encryption techniques (cf. Section 2.2.4).

Scramble! implements a user-friendly access control enforcement, by making the decryption transparent to the user. In other words, the application will parse the user's queried page and will only show the decrypted data that the user has access to. By doing a client side management of access rights, the user will be given full control on the enforcement of her privacy preferences.

We use OpenPGP standard as the encryption mechanism, which provides us with a nice PKI infrastructure and key management model, allowing us also to broadcast encryption for multiple recipients. By using GnuPG we can also make anonymous recipient encryption, although it will be still vulnerable for active attacks. Apart from dumping large amount of encrypted data, it is also possible to dump to tinyurl modules, which will post the encrypted data into a third server. In this way, the problem of the large ciphertext size that currently grows linear with the number of users that are granted access is minimised.

We have extended the plug-in to use a Java based embedded OpenPGP implementation, using the Bouncy Castle Open Source library. In this way, we provide not only a GnuPG dependent implementation, but also a more user-friendly extension version which has a built-in implementation of the OpenPGP standard. This provides users with a plug and play extension that offers the OpenPGP standard PKI infrastructure and full access control enforcement mechanisms to be used on

---

<sup>3</sup> <http://tinyurl.com/scrambleit>



social network sites. Due to the general implementation mode, the extension can also be used on a broader range of Web 2.0 applications, such as e-mail and blogs. Scramble! has been released as an Open Source Firefox application, under the EPL license.

Luo, Xie, and Hengartner provide a similar program called FaceCloak [LXH09]. Their strong point is that they do not post encryptions or tiny urls to the social network but fake information that looks as expected to the social networks. They encrypt the real content and store it on a third party server. However, in their case, they are using a symmetric key for this encryption which they then distribute, e.g., by e-mail to the intended recipient and hence users have to manage these keys by themselves. Here, our solution is much more scalable and indeed because of the use of public key encryption and PGP server, users do not have worry about the management of keys.

### ***24.2.3 Privacy-Awareness Support for Forum Users: Personal Data MOD***

When interacting with others on the Internet, users share a lot of personal data with a potentially large but “invisible” audience. An important issue is maintaining control over personal data and therefore, users need first to be aware to whom they are disclosing which data.

There are tools available that give users feedback about their IP address, location, browser etc. and that can be integrated into websites (e.g., [Mof10]). However, showing users their IP address does not mean that they know what this means and who else may see this information. Therefore, we have developed a tool for the users’ privacy awareness in a comprehensive way, i.e., a tool that informs them which explicitly and implicitly disclosed data are visible to whom. For concreteness, we have chosen to do this for the popular phpBB forum software, which is available with a copyleft license and is developed and supported by an Open Source community<sup>4</sup>. Thus, the objective of the *Personal Data MOD*<sup>5</sup> that we developed is to provide information about visibility of personal data to phpBB forum users and thereby supporting these users’ privacy awareness.

Forum users get displayed Personal Data MOD on top of the forum (see Fig. 24.1, or Fig. 24.3 for an early version). On the left side, a user is reminded about personal data from her profile and its visibility. The user is also informed about additional information which is automatically transmitted to the forum provider when visiting the forum. On the right side, a user is notified about the visibility of her latest actions and she also learns that the forum does not “forget” old actions, but that everything

---

<sup>4</sup> <http://www.phpbb.com/>

<sup>5</sup> MOD is the abbreviation for modification, a usual concept for extension of original phpBB software.

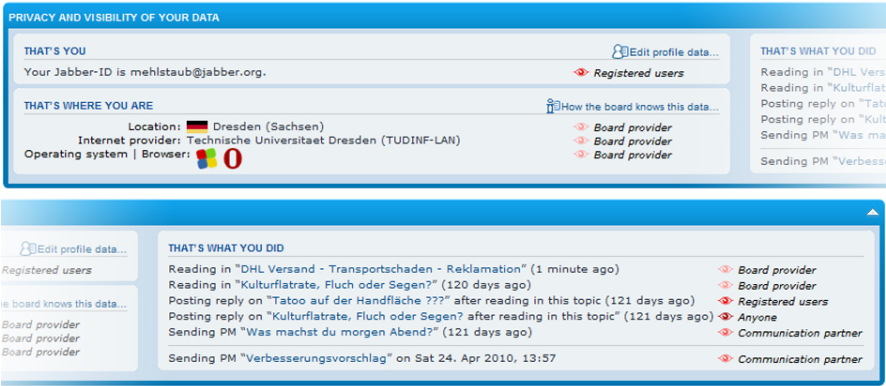


Fig. 24.1: User interface of Personal Data MOD.

is logged and can be looked up even after a longer time period. Personal Data MOD distinguishes four visibility classes for user’s personal data (see Fig. 24.2).

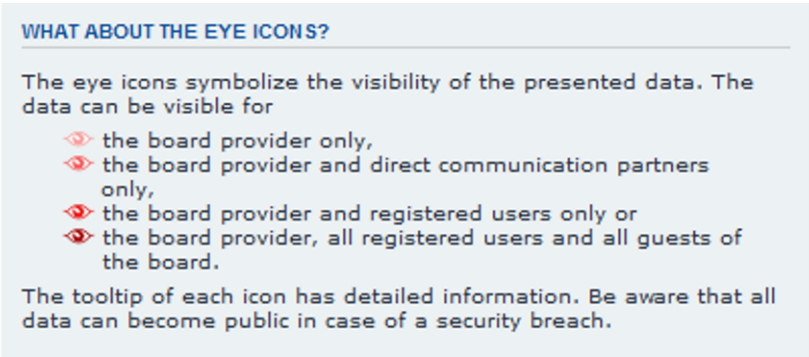


Fig. 24.2: Eye icons in different shades of red representing four visibility classes for personal data.

If a (privacy-aware) user visits the forum with Personal Data MOD via anonymising services such as AN.ON or TOR, Personal Data MOD provides feedback that the anonymising service is working correctly by indicating a location, browser and operating system which should not match with the user’s actual system.

#### **24.2.4 Privacy-Enhancing Selective Access Control for Forums**

To the best of our knowledge, no Open Source tools are available that allow forum users to specify the audiences of their posts. Therefore, we developed a phpBB extension that upgrades the access control features of the phpBB forum software so that users, instead of administrators, can define who should have access to their own contributions (e.g., thread, forum post) [PBP10, Pri10].

Since in a forum users do not necessarily know each other by name, the access control setting is done based on the other users' properties (e.g., *is over 18* or *lives in Dresden*). With the developed extension for the Open Source phpBB forum software, the user as originator is able to specify access control policies for her contribution.

The phpBB extension modifies and upgrades the original access control features of the forum, so that they work together with the access control components that were developed in the project PRIME. These components encompass:

1. creating and editing access control policies,
2. using anonymous credentials, and
3. checking access control rights.

In a forum with such extended access control features, each user is allowed to specify which properties someone has to possess in order to access the user's contribution. It is not only possible to set access control policies for the whole forum or topics, but also on a more fine grained level for threads and even single posts (cf. Fig. 24.3). This means, the user is able to define a particular audience for each single contribution and, thus, privacy-enhancing identity management is realised by audience segregation based on the properties of the audience. Technically, the process of creating a new resource (e.g., a thread) includes the originator of that resource receiving the corresponding credential (e.g., cred:Owner-Thread-ID). Further, a set of default access control policies is created, which ensure that only administrators who show an administrator credential or moderators who possess a moderator credential gain the required access needed to fulfil their roles. The owner of a resource possessing the owner credential always has access to that resource and can modify the access control policies to, e.g., also allow users who live in Dresden and who can show a LivesInDresden credential read and write access to the resource.

### **24.3 Duddle – Privacy-enhanced Web 2.0 Event Scheduling**

A number of websites have made scheduling any kind of event or running a poll much simpler. Every one can just set up a poll or suggested times and dates for a meeting, send the url to the people who should participate, and then, once all people have entered their preferences, schedule the meeting or read the results of the poll. Current implementations for event scheduling are available as stand-alone



Fig. 24.3: phpBB forum with extension for privacy-enhanced access control (note: access to second post is denied).

applications [Näf10, FS10, Pro10, Sol10, Pen08] and extensions to other Web 2.0 applications (e.g., in wikis or Groupware) [Tsa10, ope10, egr10].

As in most Web 2.0 applications, privacy is only a secondary goal. The fact that everybody may create polls, cast votes to existing polls, see results of other polls, and even revise casted votes of running polls make the application easy to use but also eliminates security and privacy. When participating in a poll, one has to share personal information with the server, the other participants, and even with the whole world.

Some of the applications recognised that users demand some privacy. Therefore, the possibility to create “hidden polls”, in which only the administrator can see all votes, is offered in some solutions. To enhance security, it is possible to allow voting and modification of votes, with login/password only. However, no application tries to overcome the trust in the application provider or the poll initiator.

Applications that especially target privacy and security requirements are special e-voting applications [Adi10, Adi08, CCM08]. However, these are not fully web-based and therefore cannot be considered as an easy to use Web 2.0 application.

PrimeLife set out to demonstrate that one can indeed schedule a meeting or run a poll in a more privacy preserving manner. The result of this effort is the *Dudle* [KB09, Kel11] Web 2.0 application, which is available from PrimeLife’s website<sup>6</sup> (Fig. 24.4).

<sup>6</sup> <http://www.primelife.eu/>

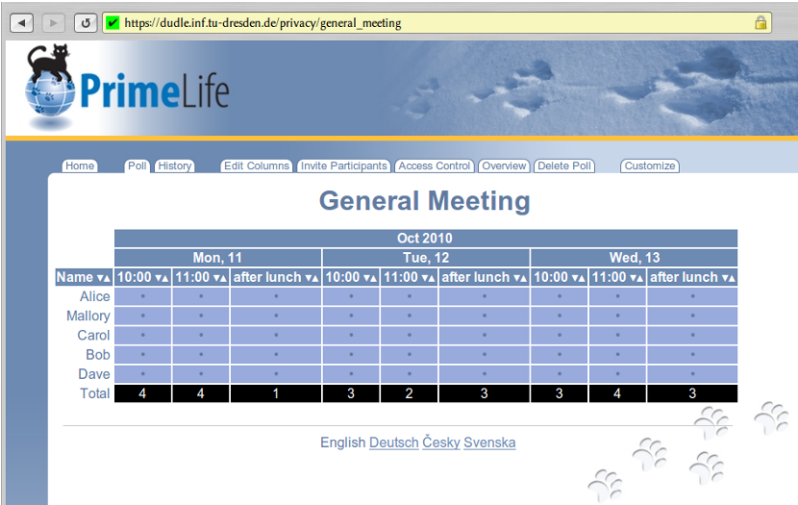


Fig. 24.4: Screenshot of Dudle. The single votes are encrypted in the browser using JavaScript.

It allows users to create polls easily and to set a number of privacy preferences. In particular, participants can submit their entries in encrypted form and the matching is done in such away that the server as well as all the participants only learn the result. This is comparable to the e-voting applications mentioned, but, due to dropping the requirements of external tool installation, it is possible to arrange and perform these multilateral secure polls within a web browser, which makes it easy to use, even for average users.

## 24.4 The Privacy Dashboard

The Privacy Dashboard<sup>7</sup>, developed within the PrimeLife project, is an extension for the Firefox browser that enables you to see some of the practices that websites are using, e.g., whether they include third party content, perhaps with lasting cookies that can track you across the Web, or are using a variety of other techniques.

The Dashboard collects information about the current website as pages load. This is presented by an icon that appears on the browser’s navigation toolbar next to the location field. The icon displays one of three aspects: a happy face, a thoughtful face and an indignant face. This is based upon rules of thumb that classify the website. The indignant face is shown if the site uses external third party HTTP cookies or external third party Flash cookies. The thoughtful face appears if the site has lasting HTTP cookies, Flash cookies or external third party content, and lacks a link to a

<sup>7</sup> <http://www.primelife.eu/results/opensource/76-dashboard>



Fig. 24.5: Screenshot of the Privacy Dashboard.

machine readable (P3P) privacy policy. Otherwise the happy face appears. These rules of thumb are to some extent arbitrary, and simply intended to draw the user's attention to the data collected.

The first time you visit a website, the Privacy Dashboard displays a privacy alert in a notification bar at the top of the page. This is the same bar used by Firefox to ask users for permission to save their user ID and password for the site. The notification bar does not appear if the site is classified with the happy face. You are invited to choose between 'accept always' (i.e., don't bother me again for this site), 'protect me', or 'tell me more'. The 'protect me' button ensures that for subsequent loads, scripting is disabled along with cookies and third party content. The 'tell me more' button displays the Privacy Dashboard dialogue window (see Fig. 24.5). The dialogue can also be displayed at any time by clicking on the Privacy Dashboard icon on the navigation toolbar.

The Privacy Dashboard dialogue has five tabs labelled “Data Track”, “Location”, “Current Website”, “Share Findings” and “About”. By default it opens with the current website tab. This shows information about the current website, your preferences for this site, and some buttons for checking the website with third party tools that, if clicked, open up in a new browser tab. The buttons cover Norton SafeWeb, Free Trust Seal, and TRUSTe.

The information shown for the site covers HTTP cookies, Flash cookies (Flash Local Shared Objects), third party content, DOM storage, geolocation, HTML5 pings, invisible images and suspicious URLs indicating the possible use of web-bugs (tracking devices). Cookies are classified according to whether they are retained beyond the current browser session, and whether they are used for this site, an internal third party site (one with a common base domain) or are for external third party sites.

The Data Track tab in the Privacy Dashboard dialogue (cf. Section 13.4) allows you to query the database of information the extension collects on each site you visit. You select from a drop-down list of queries together with a text box for typing in the domain name for a website, or a datum name or value. The queries include:

- Which data has been sent to a given website?
- Which sites has a given datum value been sent to?
- Which sites has a given datum name been sent to?
- Which sites use long lasting cookies?
- Which sites use session cookies?
- Which sites use Flash cookies?
- Which sites use DOM storage?
- Which sites are third parties?
- Which internal third parties are used by a given site?
- What cookies are used by a given site?
- Which sites use invisible images?
- Which sites use HTML5 pings?
- Which sites offer P3P policies?
- Which sites have you given access to your geographic location?

The Dashboard allows you to set personal privacy preferences on a site by site basis. The preferences are available on two levels: simple and advanced, offering a choice between three predefined levels of privacy (carefree, thoughtful and paranoid), or detailed control over a range of settings:

- Never block content from this site.
- Block external third parties.
- Block external third party cookies.
- Block all lasting cookies.
- Clear Flash cookies.
- Disable web page scripting.
- Disable access to your geolocation.
- Disable HTML5 pings.

- Don't send HTTP referrer header.
- Disable access to DOM storage.

The Firefox extension is able to implement these by directly intercepting and blocking HTTP requests, or by setting browser options.

The latter is imperfect since the option to disable scripting applies to all new pages and not just to the current tab. The extension does its best to limit changes to browser-wide options to the time the page is being loaded, but if several pages are being loaded concurrently on different tabs, then problems may well arise. Hopefully, this problem will be resolved by browser vendors offering more fine grained options that can be set on a per tab or per website basis.

The Adobe Flash plug-in is ubiquitous and installed on pretty much all web browsers. It runs in isolation from the rest of the web browser and as such makes it impractical for the Privacy Dashboard to intercept HTTP requests and to set Flash specific options. The extension is, however, able to access the local file system to examine and when requested to delete the files used for Flash Local Shared Objects.

The Privacy Dashboard also improves upon the browser's built-in support, making it easier to track and revoke which sites you have told Firefox to provide your geolocation to. If you are on a WiFi connection, you can check to see just where Google thinks you are based upon your WiFi neighbourhood.

The data collected by the Privacy Dashboard as you browse gives a view about a small part of the Web. By pooling data from many users, it will be possible to build up a much more detailed picture of how sites are tracking users. To this end, the Privacy Dashboard allows you to choose to share your findings with others. The information uploaded is limited to data about the site and its relationship to third party sites, and avoids any information that could be used to identify you. You can determine the server the uploads are made to, along with the frequency of the updates.

To encourage users to share their data, the Privacy Dashboard invites users to opt in when running the Dashboard for the first time. Thereafter, users can review and change their sharing preferences on the "Share Findings" tab on the Privacy Dashboard user interface. Servers that pool the data should avoid logging the client's IP address, time of upload, and the set of sites covered. This should be made clear in the server's privacy policy. If you are at all concerned, you can of course set your sharing preferences to use an anonymising proxy for your uploads.

There are a number of other Firefox extensions related to privacy, e.g., Adblock Plus, NoScript and BetterPrivacy. These seek to block out web page ads, to disable scripting or to offer greater control over cookies and other tracking devices. The Privacy Dashboard also does that and adds the means for users to gain greater visibility into how sites are tracking them, and the means to query this data, as well as to contribute to a broader understanding of tracking across the Web.



## 24.5 Privacy in Databases

Today, databases are often out-sourced to a database provider and sometimes also distributed over several database providers. PrimeLife has considered such scenarios and studied how sensitive personal data can be protected in such databases. In particular, we have designed and implemented two tools: Pri-Views and OverEncrypt.

### 24.5.1 *Pri-Views – Protecting Sensitive Values by Fragmentation*

When considering typical scenarios where databases are outsourced to a separate provider, one finds two important requirements: 1) the need to integrate the services of database providers that do not belong to the same organisation and 2) the presence of a variety of platforms, with an increase in the number and availability of devices that have access to a network connection, together with the presence of powerful servers offering significant computational and storage resources. The first aspect forces the requirement to specify security functions limiting access to the information stored in the databases. The second aspect instead forces an environment where the data and computational tasks are carefully balanced between the lightweight device and a powerful remote server. The two aspects are strictly related, since the servers are typically owned by service providers offering levels of cost, availability, reliability, and flexibility difficult to obtain from in-house operations. In the literature, this problem has been addressed by combining fragmentation and encryption, thus splitting sensitive information among two or more servers and encrypting information whenever necessary [CDCdVF<sup>+</sup>10].

Our contribution (Pri-Views) is a different solution to the problem. Pri-Views departs from encryption, thus freeing both the owner and the clients from the burden of key management. In exchange, we assume that the owner, while outsourcing the major portion of the data to one or more external servers, is willing to locally store a limited amount of data. The owner-side storage, being under the owner control, is assumed to be maintained in a trusted environment. The main observation behind our approach is that often it is the association between data that is sensitive, in contrast to the individual data items themselves. As with recent solutions, we therefore exploit data fragmentation to break sensitive associations; but, in contrast to them, we assume the use of fragmentation only. Basically, the owner maintains a small portion of the data, just enough to protect sensitive values or their associations.

Pri-Views offers a prototype that is mainly used to test the greedy algorithm designed to solve the problem of fragmenting data to protect sensitive associations, while limiting the data owner workload. Pri-Views takes as input a relation table and produces two views (vertical fragments) over it: one to be stored at the external service provider, and one to be directly managed by the data owner. The tool is composed of two applications: the first implements the proposed greedy algorithm

(developed in C++), while the second realises its Graphical User Interface (developed in Java).

Through the GUI, it is possible to define a data collection, characterised by a set of attributes, and a set of constraints on the joint visibility of the data in the collection. The prototype produces a fragmentation that satisfies the constraints while minimising the storage and/or computational workload for the data owner.

There are several Open Source tools that support the design of relational database schemas. For instance, SQL Workbench and SQL Power Architect permit the graphical design of relational database schemas. Most Open Source DBMSs are integrated with design tools, like pgAdmin for Postgres. The specific fragmentation design problem supported by Pri-Views is not available in these systems.

### ***24.5.2 Over-Encrypt***

Over-Encrypt is a client-server web application that provides data sharing capabilities in an outsourcing scenario where the storage service provider is trusted neither for data confidentiality nor for enforcing access control functionalities. The strong points of our solution lie in the scalability and efficiency of the data outsourcing mechanism, and in the decentralised management of access control policies and their evolution. Our approach supports the user in the specification of access restrictions to resources she wishes to share, via an external service provider, with a desired group of other users. Our proposal guarantees that only users in the specified group will be able to access the resources, which remain confidential to all the other parties, including the service itself. Scalability, efficiency and evolution of access control policies are the strong points of the proposed solution.

For our prototype, we chose Java (JDK 1.6.0) as software platform to develop the application server cooperating with an Apache Tomcat web server and a PostgreSQL database server. At the client side, we developed a Mozilla Firefox extension, with a binding to binary libraries written in C++, for the realisation of the cryptographic primitives.

The target audience for our prototype are developers who want to use state-of-the-art mechanisms to enforce access control policies over resources in a distributed environment.

A family of tools that offers a service with some similarity to Over-Encrypt is represented by the tools for on-the-fly-encryption (OTFE). There are more than a dozen Open Source tools that support this service. Among them, TrueCrypt is the most well known. These tools support the encryption of the content of the file system, offering to the user transparent access to the file system, which is stored encrypted on the disk: if the passphrase is not provided at the start of the system, the file system content is not accessible. There are two crucial aspects that distinguish Over-Encrypt from these systems. (1) OTFE tools do not typically support access control enforcement and only assume a single owner having complete access to the data; a single key is sufficient for each protected file system. (2) OTFE tools assume

that the storage device is the local disk, whereas Over-Encrypt assumes the use of a remote storage provider.

## 24.6 Anonymous Credentials

We have argued in Chapter 5 of this book that private credentials (or privacy-enhancing PKIs) are a fundamental building block to achieving privacy in authentication. Essentially, users are issued attribute certificates from different organisations and can then later selectively reveal these attributes to a relying party without revealing any of the other attributes.

Further, in Section 18 we described our vision on privacy-preserving access control systems that are based on such private credentials (cf. Section 18.1) and how they can be implemented on the basis of standardised technologies (cf. Section 18.4).

The following two subsections elaborate on our Open Source implementations of private credential systems on the one hand and components for building privacy-preserving credential-based access control systems on the other.

### 24.6.1 Identity Mixer Crypto Library

Identity mixer (Idemix) is an implementation in Java of a private credential system based on Camenisch-Lysyanskaya scheme [CL01]. More precisely, it is a library that allows one to issue credentials and prove ownership of credentials. Thereby, the library also supports other cryptographic objects such as pseudonyms, encryptions of attributes and commitments to attributes.

To orchestrate all these cryptographic objects, we have developed a specification language for the issuing protocol and also for the credential presentation protocols. [Fig. 24.6](#) provides an example of this specification language for the credential specification protocol (also called proof protocol). In this example, a user proves possession of three different credentials, each of which has the same value for the attribute `LastName`. This value is further encrypted under a public key `PublicKey1`.

The library and documentation are available from the PrimeLife webpage. We refer to that documentation for a detailed description of the architecture, all specification languages, and details for the underlying cryptographic protocols.

```

Declaration{ id1:unrevealed:string; id2:unrevealed:string;
             id3:unrevealed:int; id4:unrevealed:enum;
             id5:revealed:string; id6:unrevealed:enum }
ProvenStatements{
  Credentials{
    randName1:http://www.ch.ch/passport/chPassport10.xml =
      { FirstName:id1, LastName:id2, CivilStatus:id4 }
    randName2:http://www.ibm.com/employee/employeeCred.xml =
      { LastName:id2, Position:id5, YearsOfEmployment:id3 }
    randName3:http://www.ch.ch/health/healthCred10.xml =
      { FirstName:id1, LastName:id2, Diet:id6 } }
  Enums{
    randName1:CivilStatus = or[Marriage, Widowed]
    randName3:Diet = or[Diabetes, Lactose-Intolerance] }
  Commitments{ randCommName1 = {id1,id2} }
  Representations{ randRepName = {id5,id2; base1,base2} }
  Pseudonyms{ randNymName; http://www.ibm.com/employee/ }
  VerifiableEncryptions{ {PublicKey1, Label, id2} }
  Message { randMsgName = "Data to be used only for ..." }
}

```

Fig. 24.6: Example proof specification using a Swiss passport, an IBM employee credential, and a Swiss health credential.

### 24.6.2 Components for a Privacy-Preserving Access Control System

The PrimeLife Policy Engine (cf. Section 20) is an implementation of the policy concepts developed throughout the project. Those concepts comprise access control as well as usage control (also called data-handling) aspects. Although the engine itself is not available as an Open Source implementation, the components that concern the credential-based access control aspects are provided as Open Source. Those components constitute the major building blocks necessary for developing a stand-alone privacy-preserving credential-based access control system. To understand which components are provided in particular, consider [Fig. 24.7](#) that illustrates how a privacy-preserving access control transaction takes place as well as the following description of the Figure.

A transaction involves three kinds of entities: users, servers (also called service providers), and issuers. Initially, a user contacts a server to request access to a resource she is interested in (1). Having received the request, the server responds with the credential-based access control policy applicable for the resource (2). The applicable policy may be a composition (1a) of multiple policies the server holds. Upon receiving the policy, the user's system evaluates which claims she can derive from her available credentials that fulfil the given policy (2a). The favoured claim is then chosen by the user interactively or automatically (2b). If the user wants to proceed, evidence for the chosen claim is generated by a plug-in specific to the respective credential technology (2c) and sent, together with the claim and the attributes to

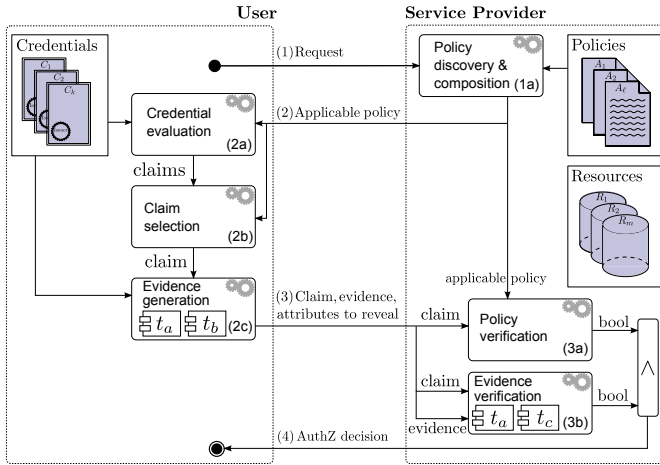


Fig. 24.7: Decision rendering in our system model.

reveal, to the server (3). Finally, the server verifies whether the policy is implied by the claim (3a) and if the evidence supports the validity of the claim (3b). If so, access to the resource is granted (4).

The Open Source components that we provide are capable of performing steps (2a), (2c), (3a) and (3b). For performing step (2b), the *Send Data* dialogue that is described in Section 14 could be employed. For implementing an entire privacy-preserving access control transaction as described above, the provided open-source components have to be integrated with an existing access control system and an appropriate messaging standard. In Section 18.4, we elaborate on how this can be done for XACML and SAML and we plan to provide an implementation of those concepts at the end of the project, i.e., an implementation capable of performing an entire privacy-preserving access control transaction.

## 24.7 Conclusion

The sections above illustrate that within and outside the PrimeLife project, a vast amount of initiatives in Open Source related to privacy and identity management are being developed. With time, some of these will grow and become de facto standards and tools, while some will perish. This selection process will, contrary to what one might expect, not necessarily result in survival of the fittest. One of the problems in an Open Source setting is the inability to develop sustainable packages: Open Source products are developed, released and then left as is without active maintainers. It may be naive to assume that Open Source products prove themselves in such a case, and that the best solutions will stand out and survive because of their technical superiority or user-friendliness. To stay worthwhile, an Open Source project should

grow: attract new developers, new community members, and new money. It should scale in functionality, which also implies more complexity. Therefore, an active advertisement and uptake by large players is needed to bring even a superior product to the large masses. One example of this is the OpenOffice productivity suite, that developed a Strategic Marketing Plan within the OpenOffice.org Marketing Project<sup>8</sup>.

General recommendations regarding Open Source can be found in the 2020 FLOSS Roadmap.<sup>9</sup> Some of these recommendations can be made more specific for the privacy and identity management field. Regarding openness and freedom in ICT infrastructures, the recommendations are that network neutrality should be protected with a legal framework, and by developing decentralised, user-controlled, free software-based web services for all essential social or collaborative applications. This is necessary to keep users' data under users' control, but as well as to ensure that network anonymity technology is not blocked. Furthermore, the FLOSS Roadmap mentions that government entities should actively seek FLOSS-based solutions as much as possible. Particularly in the privacy and identity management area, this is a good strategy: as government entities typically deal with the identities and sensitive data of their citizens, they can be the first to set the example and introduce Open Source solutions to society. Open Source software that could help to protect the privacy of citizens should therefore be supported, used and integrated by government institutions.

Apart from these general considerations, specific technologies in privacy and identity management remain insufficiently supported in Open Source, if supported at all.

First of all, one essential part that is underdeveloped at the moment is a widespread framework into which Open Source privacy and identity management modules can be plugged. A zoo of small solutions, each for one specific application or use, each with its own settings and installation defaults, can only be maintained by ICT-savvy people. Typical tasks of such a framework are the storage of the user's privacy preferences (privacy policy), and the validation of requests against those preferences. The work within PrimeLife in the area of policy languages certainly establishes a good starting point for the development of such a framework. Another component of such a framework could be the Identity Metasystem [CJ06], taking care of the identity management of users, which should – in theory – be agnostic of specific identification technology. A considerable amount of technology that implements the Identity Metasystem is Open Source already, including identity providers as well as relying party components. The Higgins framework and Microsoft code name 'Geneva' are efforts in this area. The development of U-Prove<sup>10</sup> by Microsoft and the IBM's Identity Mixer (Idemix) are available from PrimeLife's website. Recent work, seeded by Southworks, on bridging OpenID and WS-Federation,<sup>11</sup> indicates that more technologies are integrated into the idea of the Identity Metasystem.

<sup>8</sup> <http://marketing.openoffice.org/>

<sup>9</sup> FLOSS – Free Libre and Open Source Software; <http://www.2020flossroadmap.org/>

<sup>10</sup> <https://connect.microsoft.com/content/content.aspx?contentid=12505&siteid=642>

<sup>11</sup> <https://github.com/southworks/protocol-bridge-claims-provider>

Despite this promising progress in the Identity Management part of such frameworks, advanced privacy technology such as the PrimeLife policy language and its enforcement engine are yet to be integrated.

Secondly, awareness tools and (related) data analysis tools for users are not thriving in Open Source. The main efforts are seen in browsers implementing a primitive form of data minimisation (restrictions on cookies, 'anonymous' browsing, etc.), and some support for privacy policy languages. Also, classical virus scanners, firewalls, disk cleaners and anti-spyware will protect against unauthorised collection of data, but fail to assist the user with advice regarding the data that she wants to release voluntarily. The Privacy Dashboard developed within PrimeLife is one effort in that direction, aimed at providing the user with an overview of the data she released.

Furthermore, most of the tools described above work well in a PC setting, but this is a platform that is losing ground at a rapid pace. A majority of the people born today will not use a classic PC to access the Internet in their daily routine. Devices such as smartphones, game consoles, TV sets, tablet PCs and even ordinary household appliances will try to provide seamless network access everywhere and at any time. "According to Gartner's PC installed base forecast, the total number of PCs in use will reach 1.78 billion units in 2013. By 2013, the combined installed base of smartphones and browser-equipped enhanced phones will exceed 1.82 billion units and will be greater than the installed base for PCs thereafter."<sup>12</sup> This opens up a range of additional concerns:

- Even if these devices will have the same computational power and software installed on them in the near future, they are present and being used at this moment. In the meantime, very sensitive information about the users can be collected. In any case, porting Open Source solutions to these devices – if at all possible – will take some time and effort.
- An additional problem is that some devices come with closed operating systems, for which the integration of Open Source poses a threat, especially when the related Open Source licenses behave in a viral way.
- New software challenges will arise when devices do not support a classical user interface, or feature peculiar hardware components. Again, additional efforts will be needed to adjust (and not only port) Open Source solutions to these devices. Moreover, mobile phones initiate fundamentally different uses than PCs due to their portability and their enhanced sensors. One obvious example is location, determined by GPS, WiFi or GSM cell. This triggers additional privacy challenges, even for non-users.<sup>13</sup>

<sup>12</sup> Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond; <http://www.gartner.com/it/page.jsp?id=1278413>

<sup>13</sup> Four Billion Little Brothers?: Privacy, Mobile Phones, and Ubiquitous Data Collection; <http://cacm.acm.org/magazines/2009/11/48446-four-billion-little-brothers/fulltext>

In any case, we see that currently, these new devices are in their infancy with respect to privacy technology, and that psychological barriers that existed in PC users are easily discarded.

Finally, at the network layer, new technology will have to be developed to enable anonymity. Existing Open Source implementations such as TOR and the related JonDo, based on the AN.ON project, remain slow for common use, and even for ordinary operations that require limited bandwidth. This is a huge problem, considering the continuous move towards faster networks necessary for online gaming, Video-on-Demand, IPTV, etc. Related to this challenge is the move towards cloud computing. Obviously, this can generate new privacy problems, especially in the case of SaaS (software as a service), which is in fact the recentralisation of data and processes, from end-user devices to entities, possibly operated by a single controller. On the other hand, the cloud can also be used in favour of privacy. In essence, TOR is a kind of cloud computing, and so is the Diaspora project,<sup>14</sup> in which a distributed, privacy-aware, personally controlled Open Source social network is being developed.

Summarising, additional funding and promotion of Open Source products for privacy and identity management should be aimed at:

- identifying and marketing specific, technically superior Open Source products actively,
- establishing an interoperable framework in which Open Source modules for privacy and identity management can be developed and plugged,
- funding the creation of advanced user awareness tools in Open Source,
- Open Source for end-user devices other than the classic PC,
- new anonymity networking software for broadband traffic,
- Open Source initiatives that leverage the possibilities of cloud computing for privacy technology.

---

<sup>14</sup> Diaspora: The privacy aware, personally controlled, do-it-all, Open Source social network.  
<http://www.joinindiaspora.com/>





## Chapter 25

# Contributions to Standardisation

Hans Hedbom, Jan Schallaböck, Rigo Wenning, and Marit Hansen

### 25.1 Introduction

Standardisation has many goals and facets: Standards are used for consumer protection to achieve a minimum quality of certain products and services. Standards lead to lower cost because of a unified higher volume market. Standards also support interoperability that is vitally needed in ICT.

The ICT landscape is characterised by an extensive division of labour between specialists. The device driver programmers rely on the information that the device manufacturers give them. The operating system developers rely on information and interfaces that are provided by the device drivers and by the CPU instruction set. Application developers rely on the interfaces from the operating system and web developers rely on the interfaces the Web provides. This means that ICT has a much larger need for agreed information that leads to interoperability. In short, ICT needs many more standards than the rest of the industry.

But the function of standards in ICT goes far beyond pure interoperability. A new set of interfaces is sometimes the way to open an entire new world thus creating new markets. For instance, it took new standards to bring the Web to mobile devices thus creating a huge new market for applications and commerce.

Quite often, the idea for such new markets comes out of research. However, researchers are usually not taking the pains to actually create the market. Mostly they are satisfied to show that it theoretically should work and perhaps provide a demonstrator to showcase what it could look like. The European Commission realised this gap and consequently has recently put a lot of emphasis on the relation between research and standardisation.<sup>1</sup>

Traditional industry standardisation is rather directed on achieving agreement among several vendors whose products have converged sufficiently to formalise the common understanding of how things should be done. Further, standardisation is used by public authorities to achieve goals of consumer protection.

---

<sup>1</sup> For further information on research and standardisation see <http://copras.org/>.

In both fields, PrimeLife has developed activities and has generated impact in the emerging markets concerning privacy and identity management. PrimeLife's part in ISO<sup>2</sup> standardisation has focused on high level framework and platform specifications that contain requirements on privacy-respecting software design. The W3C<sup>3</sup> work has concentrated around enduring the dialogue between web developers, browser makers and researchers, understanding privacy issues of the Web, presenting possible solutions and searching for a possible consensus with the web community. Finally, PrimeLife used the opportunity to offer drafts to the Network Working Group of the IETF<sup>4</sup>.

In the following, this section gives an overview of PrimeLife's approach to giving input to the ISO/IEC standardisation (cf. Section 25.2), the project's collaboration with the W3C (cf. Section 25.3) and some results in the cooperation with the IETF (cf. Section 25.4).

## 25.2 Standardisation in ISO/IEC JTC 1/SC 27/WG 5

In ISO, the joint technical committee ISO/IEC JTC 1/SC 27 is in charge of standardising security standards for information systems. Among other things, they are behind the 27000 series on information security management systems. Within SC 27 the working group 5 (WG 5) is responsible for standards within the identity management and privacy area.

Early on, PrimeLife established a cooperation with WG 5 in the form of a liaison agreement with the group. The reason for the liaison is that WG 5 is working on a number of standards that have commonalities with the aims and the scope of the PrimeLife project and we wanted to be able to influence these standards and to contribute with our knowledge and findings in the standardisation process. The contributions of PrimeLife have been very well accepted by WG 5 and we believe that we have had mutual benefit from the cooperation. Even though the whole spectrum of the standards within WG 5 is of interest, there are three projects that lie close to the work going on in PrimeLife and we have therefore decided to concentrate our contributions to these standards.

The projects concerned are the 24760 "A Framework for Identity Management" standard, the 29100 "Privacy Framework" standard and the 29101 "Privacy Reference Architecture" standard. Most of the contributions have been in the form of discussions on work group meetings and comments on standard drafts; however, there are some areas where PrimeLife has made very significant impact. The remainder of the subsection will discuss specifically PrimeLife's input to the Framework for Identity Management and the Privacy Reference Architecture.

---

<sup>2</sup> ISO stands for International Organization for Standardization. In ICT its work is often aligned with the standardisation within IEC (International Electrotechnical Commission), cf. <http://www.iso.org/>.

<sup>3</sup> World Wide Web Consortium, cf. <http://w3.org/>.

<sup>4</sup> Internet Engineering Task Force, cf. <http://www.ietf.org/>.

### ***25.2.1 ISO 24760 – Framework for Identity Management***

ISO 24760 aims at describing a framework for identity management and defining its components. The standard presents terminology, concepts, identity life cycle and best practices within the identity management area. It started out as a monolithic standard, but after suggestions from PrimeLife and other contributors, it was divided into three parts. The biggest issue within the standard has been around terminology and the interpretations of the different terms. There were also some discussions on the format of the descriptions of the different terms. PrimeLife suggested a total make-over of the structure and format of the terminology and as a result of this, one employee at one of our partners became the co-editor of the standard.

Identity is an important and ambiguous concept in identity management. The understanding of the term (and the implications of that understanding) ranges from a collection of attributes associated with an individual to a collection of attributes making an individual unique. In the realm of natural or legal persons, it is easy to argue that an identity is a collection of attributes associated with an individual.

However, if the identity concept is pushed into the realm of objects, the understanding or the limits of the concept becomes problematic. Potentially, one could argue that one unit of data would be an identity or that everything is an identity if an identity is defined as a collection of attributes associated with an object. A consequence of this is then that every computer system is an identity management system, which is not in line with the understanding of the experts in the field and could also make the concept of identity essentially useless since nothing exists that is not an identity.

On the other hand, requiring that an identity always uniquely identifies the entity blurs the difference between identity and identifiers. More or less, this understanding makes it pointless to allow a user to have multiple identities in the system and potentially creates large privacy problems. As a consequence, one of the biggest issues regarding the terminology has been the concept of identity including terms like identifier and partial identity. The problem with partial identity is that the concept is rather new and not used that much outside of research circles.

Some of the attending experts thought that it could be hard to push it into an industrial setting even if they do agree with and understand the concept. In the terminology discussions, PrimeLife has provided its view of the concepts. PrimeLife also contributed in making the document consistent and in advocating the users' view and tried to gear the standard into a more user-centric model by providing the experience gained and discussions held during the project.

### ***25.2.2 Introducing Privacy Protection Goals to ISO 29101 Privacy Reference Architecture***

IT security<sup>5</sup> and privacy protection are overlapping perspectives when implementing IT systems. They both need to be considered already at the level of developing underlying architectures.

Usually, IT security takes the perspective of an organisation, i.e. the objective is to safeguard the assets of that organisation. Here the “Classic CIA Triad”<sup>6</sup> of the IT security protection goals (confidentiality, integrity and availability) is applied as necessary for the specific context. These protection goals are useful to structure risks and countermeasures, and to set up a working Information Security Management System (ISMS).<sup>7</sup>

In contrast, privacy protection focuses on the individuals concerned, i.e., the Data Subjects. Certainly the IT security protection goals confidentiality, integrity and availability are important here, too, but they do not represent all areas that should be covered when it comes to the privacy of an individual as well as to the compliance with today’s data protection regulation.<sup>8</sup>

IT security protection goals such as confidentiality, integrity and availability may facilitate the implementation of privacy principles into an IT system, but do not suffice to cover all aspects of privacy protection. For privacy protection, these goals need to be complemented with a set of specific protection goals that also allow for the expression of mismatches and conflicts of different goals. Even with the three classical IT security protection goals, it always has to be determined how much each goal should be pursued and what balance between conflicting aspects of those goals should be achieved. With the extension to six of those high-level protection goals, potential conflicts are more visible, which is good because they have to be tackled when designing, operating and improving the IT systems. There is no “one size fits all” solution, but for each application context, individual balances and implementations have to be determined, dependent on, e.g., the sensitivity of data, the attacker model, legacy issues from already existing components of the information system, and last but not least, legal obligations.

To allow for a more holistic mapping of privacy principles, the three IT security protection goals are supplemented by three privacy-specific protection goals: transparency, unlinkability and intervenability, as explained below. A hexagon of protec-

<sup>5</sup> Note that we use the term “IT security” in its broad meaning of “information security” covering all security aspects of the full information system, regardless, whether technological components are involved or not. Among others, organisational processes, data on all kinds of media or the staff involved in data processing are part of this comprehensive approach, e.g., when analysing risks or selecting and implementing appropriate countermeasures.

<sup>6</sup> CIA stands for Confidentiality, Integrity and Availability, not for the well-known secret service.

<sup>7</sup> For more information see the standards ISO/IEC 27001 and ISO/IEC 27002 as well as ISMS in the frameworks ITIL (Information Technology Infrastructure Library) and COBIT (Control Objectives for Information and related Technology).

<sup>8</sup> Credit for the research underlying this section goes to Martin Rost and Andreas Pfitzmann, see also [RP09] and [RB11].

tion goals can be derived where each goal is countered with another one expressing dualistic aspects of the protection, see Fig. 25.1 (cf. [RB11]). All protection goals can in principle be applied both on the information itself, as well as on the processes, and technical layers. For each, the perspective of the Data Controller, the Data Subject and a third party can be adopted. Privacy protection goals help to structure risks and to define which measurements to apply.

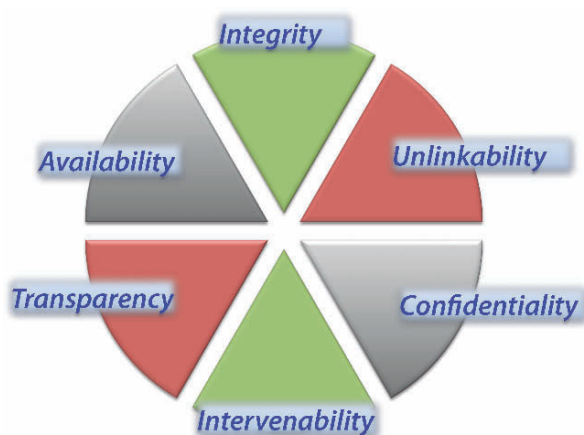


Fig. 25.1: Segments of security and privacy protection goals.

To support and develop a common understanding of the aforementioned concepts that could only be addressed briefly herein, the terms and definitions above have been submitted as a comment from PrimeLife to the drafting of ISO 29101 Privacy Reference Architecture.

In the following, the privacy-specific protection goals are explained:

*Transparency:* For all parties involved in privacy-relevant data processing<sup>9</sup> (specifically the Data Controller, Data Processor(s), Data Subjects as well as supervisory authorities), it is necessary that they are able to comprehend the legal, technical, and organisational conditions setting the scope for this processing. Examples for such a setting could be the comprehensibility of regulatory measures such as laws, contracts, or privacy policies, as well as the comprehensibility of used technologies, of organisational processes and responsibilities, of the data flow, data location, ways of transmission, further data recipients, and of potential risks to privacy. All these

<sup>9</sup> The term “privacy-relevant data processing” comprises all kinds of data processing that is or may be privacy-relevant, i.e., have some influence on the privacy of individuals. Thereby it intentionally chooses a wider approach than “processing of personal data” as it is regulated in European data protection law. The term “privacy-relevant data” encompasses at least personal data and potentially personal data.

parties should know the risks and have sufficient information on potential counter-measures as well as on their usage and their limitations.

Transparency is a necessity for important aspects of informational self-determination, such as access rights, informed consent and notification obligations of data processors. It can be achieved or enhanced by several mechanisms, such as documentation, logging, reporting, data protection management systems as well as information of and communication with the Data Subject.

*Unlinkability:* Unlinkability means that all privacy-relevant data processing is operated in such a way that the privacy-relevant data are unlinkable to any other set of privacy-relevant data outside of the domain (or the applicability of a well defined purpose), or at least that the implementation of such linking would require disproportionate efforts for the entity establishing such linkage. Unlinkability is the key element for data minimisation as well as purpose binding. Its objective is to minimise risks to the misuse of the privacy-relevant data and to prohibit or restrict profiling spanning across contexts and potentially violating the purpose limitations related to the data.

Wherever feasible, Data Controllers, Data Processors, and system developers should completely avoid or minimise as far as possible the use and possibilities for linkage of privacy-relevant data, conceivably by employing methods for keeping persons anonymous, for rendering persons anonymous (“anonymisation”), or for aliasing (“pseudonymisation”). Observability of persons and their actions as well as linkability of data to a person should be prevented as far as possible. If privacy-relevant data cannot be avoided, they should be erased as early as possible.

*Intervenability:* Intervenability aims at the provision of possibilities for Data Subjects, Data Controllers as well as supervisory authorities to intervene in all kinds of privacy-relevant data processing, where necessary. The objective is to offer corrective measures and counterbalances in processes. For Data Subjects, intervenability comprises the Data Subject rights to rectification and erasure or the right to file a claim or to raise a dispute in order to achieve remedy when undesired effects have occurred. For Data Controllers, intervenability allows them to have efficient means to control their Data Processors as well as the respective IT systems to prevent undesired effects. Examples for such means may be the ability to stop a running process to avoid further harm or allow investigation, to ensure secure erasure of data including data items stored on backup media, and manually overruling of automated decisions or applying breaking glass policies. For supervisory authorities, intervenability could consist of ordering the blocking, erasure or destruction of data, or in severe cases stopping the data processing entirely.

Intervenability can be achieved or supported by mechanisms such as the provision of a single point of contact (SPOC). Other approaches include a separation of processes, as a means to allow the system to continue to be working, even if there is the need for intervention in a specific case. The Data Subject should be offered an easy and convenient way to exercise the Data Subject rights to rectification or erasure of personal data as well as withdrawing previously given consent.

## 25.3 Web Privacy

Privacy-enhancing technologies are great consumers of access control technology. PrimeLife is in no way an exception here. The early works and research on the PrimeLife model and policy engine consequently focused on access control and how to organise it. At the same time, with some help of the European Commission, a coordination with other projects was organised. The coordination between the projects was called “PrimCluster”. Rapidly after the first PrimCluster meetings with projects SWIFT,<sup>10</sup> TAS3,<sup>11</sup> and PICOS,<sup>12</sup> it became clear that all projects were using and extending the eXtensible Access Control Language (XACML) specified by OASIS. Further inquiry in the community revealed that there were more projects beyond the ones organised in PrimCluster that had new ideas and innovative extensions concerning XACML. The topic was brought up in the Policy Languages Interest Group (PLING)<sup>13</sup> to determine interest from the industry. The response was positive. PrimeLife decided to allocate the necessary resources for a *Workshop on Access Control Application Scenarios* that would look specifically at XACML innovations and beyond. W3C organised the workshop as a standardisation workshop in November 2009 in Luxembourg.<sup>14</sup>

As the Web advances toward becoming an application development platform that addresses needs previously met by native applications, work proceeds on APIs to access information that was previously not available to Web developers. Work on Web Applications and on the Geolocation API for web sites triggered intensive privacy discussions. Device APIs providing broad availability of possibly sensitive data collected through location sensors and other facilities in a web browser is just one example of the broad new privacy challenges that the Web faces today. The privacy discussion was also brought into PrimeLife for further consideration and to consider possible solutions. The dialogue was further broadened by PrivacyOS where several stakeholders had first discussions.<sup>15</sup> All this together led to the *Workshop on Privacy for Advanced Web APIs*<sup>16</sup> in July 2010 in London to discuss the current work on the user facing side within a broader audience.

However, already at the Workshop on Privacy for Advanced Web APIs in London it became clear that access control is not enough, neither on the server side nor on the client side. While the London Workshop created a community willing to address the technical challenges of privacy in the Web context, and while they started to have

---

<sup>10</sup> “Secure Widespread Identities for Federated Telecommunications”, <http://www.ist-swift.org/>

<sup>11</sup> “Trusted Architecture for Securely Shared Services”, <http://www.tas3.eu/>

<sup>12</sup> “Privacy and Identity Management for Community Services” <http://www.picos-project.eu/>

<sup>13</sup> <http://www.w3.org/Policy/pling/>

<sup>14</sup> All the proceedings, minutes and papers are available under <http://www.w3.org/2009/policy-ws/>.

<sup>15</sup> <https://www.privacyos.eu/>

<sup>16</sup> <http://www.w3.org/2010/api-privacy-ws/>



lively discussions on a W3C hosted mailing-list,<sup>17</sup> ideas about what to do are clearly not shaped yet. W3C organised a further *Workshop on Privacy and Data Usage Control*<sup>18</sup> in October 2010 in Cambridge (MA) to encourage further discussions on the question of data usage once data has been collected. This again involved requirements and expectations from the Device API community.

In December 2010, the last of the workshops in PrimeLife's series of events dedicated to standardisation was co-organised with the Internet Architecture Board (IAB), W3C, the Internet Society (ISOC), and Massachusetts Institute of Technology (MIT): the *Workshop on Internet Privacy* in Boston.<sup>19</sup> A broader scope was chosen intentionally to discuss upcoming issues in online privacy that need to be tackled on a global scale.

In the following subsections, relevant results from the four workshops are briefly described.

### **25.3.1 Workshop on Access Control Application Scenarios**

The Workshop on Access Control Application Scenarios<sup>20</sup> attracted 20 position papers of rather diverse nature. Most of them were presented in the two day workshop in Luxembourg and the discussion converged towards four topics:

- Attributes
- Sticky Policies
- Obligations
- Credential-based Access Control

#### **25.3.1.1 Attributes**

XACML provides a framework for access control systems in heterogeneous IT landscapes. There is a protocol and some basic requirements that are common to all access control systems. But XACML does not specify the semantics of the conditions that have to be fulfilled to grant access. Those semantics are specified by the actual implementer within an existing enterprise. This means in order to expand to inter-enterprise interoperability or to widen use on an internet scale, XACML needs semantics filling out its own framework that makes access control conditions predictable and interoperable even where there was no prior agreement on the semantics of the access control conditions. University Bergamo and University Milano

---

<sup>17</sup> <http://lists.w3.org/Archives/Public/public-privacy/>

<sup>18</sup> <http://www.w3.org/2010/policy-ws/>

<sup>19</sup> <http://www.iab.org/about/workshops/privacy/>

<sup>20</sup> <http://www.w3.org/2009/policy-ws/>

contributed a paper describing extensions to XACML to make it easily deployable and suitable for open web-based systems.

The participants presented their different vocabularies during the workshop. Various areas were tackled: Apart from PrimeLife's privacy vocabulary, work on access control in social networking or attribute vocabularies for export control, geospatial data and health care data were outlined in the workshop. The chair invited all participants to contribute their semantics to the TC XACML, which could act as a clearing house for those ontologies. This way, duplication of attributes could be avoided and a cleared vocabulary could be standardised for a wider audience and to achieve some basic interoperability for web or inter-enterprise consumption.

### 25.3.1.2 Sticky Policies

Applying access control scenarios beyond the borders of a well-walled enterprise does not only raise the question about agreed and interoperable access control semantics. It also raises the question of how to make sure that all users of a data record can respect the access restrictions if this record is travelling around from service to service, across company borders or from continent to continent on the Web. One solution is known under the name "Sticky Policy."<sup>21</sup> This means that there is a persistent link between the access control information and other metadata and the record containing e.g., personal data. A parallel issue exists for Digital Rights Management (DRM). There are several co-existing possibilities to organise the "Sticky Policies", e.g., by using a binding as in XML Signature (detached and in line), possibly supported by an online data store that contains the bindings, so that the Policy Enforcement Point (PEP) could just ask there.

An additional issue came up while considering that access policies with conditions travel around. The sending service has a set of policies, but the receiving service also already has a certain set of policies (endogenous policies). In practice, those policies must be combined in order to compute a concrete result on whether access can be granted, or whether the receiving service is able to accommodate the requirements from the sending service. It quickly became clear that the combinability of policies turns into a major requirement once more complex distributed systems or ad-hoc systems are considered. There are several algorithms already available, but none of them is currently standardised. But standardisation of the algorithm of combination is needed to design policies and systems with predictable results. XACML currently provides a built in set of policy combining algorithms, but work is need to determine their suitability for this application.

---

<sup>21</sup> See also Part IV.

### 25.3.1.3 Obligations

For privacy policies, there are conditions and actions that are not tied to an access control event. For the moment, XACML has an intentionally underspecified <Obligation> element that people were using in creative ways. But this underspecification has the side effect of undermining the interoperability of such obligations. Thus, one cannot be sure whether the specified actions are actually performed by the receiving service. One of the immediate requirements was that if the receiving service does not understand the obligation, it should deny access with a feedback to the requester.

PrimeLife inherited an obligation language from the PRIME project and developed it further. This was presented at the workshop. Also, other projects presented their work on obligations. Some participants suggested the use of Semantic Web technologies and the use of the W3C Rule Interchange Format (RIF). At this early stage, it was decided that further work was needed, possibly coordinated among several projects that could lead to a suggestion for TC XACML.

### 25.3.1.4 Credential-based Access Control

Credential-based Access Control would allow for a more privacy-friendly access control system that would also be more widely usable on the Web. The aim is to prove only selected attributes as needed for the task at hand. There is already a large set of literature on capabilities, but XACML currently does not have the ability to identify the type of credential used nor to specify which credential is needed to get access to a certain resource. This is more or less a special case of the attributes topic with additional protocol issues. One way to convey the credential would be to use SAML, but SAML only allows XML Signature as a proof token.

Further steps in this direction are already undertaken and the actual PrimeLife protocol will be contributed to TC XACML to address credentials as access control conditions. But the contribution will also make XACML itself more privacy-friendly. Today, if a user hits an access controlled resource, the system simply returns this resource as restricted. The user then tries as many credentials as she has until the resource opens. The XACML 2.0 protocol has no way to tell the user which credential it requires to open the access to the desired resource. The PrimeLife extension enables the Policy Decision Point (PDP) to convey the type of credential it wants already in the response to the initial attempt to access a resource.

## 25.3.2 *Workshop on Privacy for Advanced Web APIs*

The workshop on Privacy for Advanced Web APIs served to review experiences from recent design and deployment work on device APIs, and to investigate novel strategies toward better privacy protection on the Web that are effective and lead to

benefits in the near term. It focused on work done by the W3C Geolocation Working Group, the W3C Device API and their security and privacy considerations. We already see new Appstores with applications for our mobile devices. Web applications use web technology to provide such applications for desktop and mobile devices. Questions from the Web Applications Working Group were getting even more emphasis by the W3C Technical Architecture Group's recent and future work on a Web Application Architecture.

PrimeLife came just in time to help organise this important workshop and also used it to distribute some of its results.

The two practical proposals that drew most interest and discussions were the Mozilla privacy icon approach.<sup>22</sup> and CDT's privacy rule-set idea<sup>23</sup> Both proposals received a lot of positive feedback, and questions about their viability. In addition to technical and user interface challenges, there were questions about the business incentives for browser vendors and large web providers, as one of the main obstacles for getting privacy from research and standardisation to deployment. Nevertheless, further investigation and experimentation with both approaches seems worthwhile and was encouraged.

There was agreement that it is useful to capture the best current practices gained during early implementation efforts (such as those presented during the workshop regarding the geolocation API). Furthermore, investigating how to help specification writers and implementers to systematically analyse privacy characteristics in W3C specifications was seen as a worthwhile effort. The wealth of discussions and the enthusiasm of the participants of the Workshop encouraged people to continue the dialogue in a mailing list and possible future workshops.

### ***25.3.3 Workshop on Privacy and Data Usage Control***

As a complement to the considerations on access control and also as a complement to the considerations around APIs and the new challenges for web user agents they bring, there are also back-end considerations. Service side operations raise privacy questions beyond mere database design. How would a service make sure that data are used within the boundaries of the promises that had been given to the user and maintain the boundaries even if third party services are used to fulfill the user's needs? It becomes immediately clear that the service side of things also has large implications for the user agent part of the equation. As a consequence, the Device API Working Group presented their list of requirements for privacy and looked for possible solutions.

The workshop revealed that the complexity presented by PrimeLife to the audience was not really an issue for the service oriented businesses that typically handle large amounts of data within complex systems. It was also clear that extending the

---

<sup>22</sup> <http://www.w3.org/2010/api-privacy-ws/papers/privacy-ws-22.txt>

<sup>23</sup> <http://www.w3.org/2010/api-privacy-ws/papers/privacy-ws-12.html>

complex system to the user side of things and to user agents would not work either. What could work is a set of simple semantics in the dialogue with the user and her user agent and only use the full complexity of solutions within or between privacy-enhanced services.

It became also clear that there is still a lot of education and communication needed. Developers and also sections of the software industry determining protocols and capabilities are still not sufficiently aware of the fundamental insights and goals of privacy. The translation of high level privacy goals that has last been done in the 1980ies with the OECD Guidelines and the Census Decision of the German Federal Constitutional Court and also subsequently with the Directive 95/46/EC on Data Protection into concrete and tangible hints and advises for software development on the Internet and on the Web is still missing. We do not really understand yet what the information revolution of the past 20 years has brought. We only start to realise that the old system of self-determination that tends to become a bean counting exercise is not what will help create technical remedies for our everyday life on the Web. So a new effort of translation is needed. This means philosophers, technicians and lawyers have to reconvene in discussions on what the threats really are, what goals can be set and achieved. This suggests further interdisciplinary workshops.

It is also clear that the topic that is still missing in the discourse we had is the economy of privacy. On the Web, personal data are a currency and privacy protection is swimming against the stream of the billions earned by targeted advertisement. So one of the questions that will have to be considered is: What framework will be needed to encourage investment into privacy tools rather than into lucrative tracking tools that augment the return per served ad?

### ***25.3.4 Workshop on Internet Privacy***

At the Boston Workshop on Internet Privacy in December 2010, the 60 workshop participants from enterprise, governments, civil society, academia, as well as various standardisation bodies discussed the question “How Can Technology Help to Improve Privacy on the Internet?” The objective was to explore conflicting goals of openness, privacy, economics, and security to identify a path forward for improving privacy. The discussed topics ranged widely, and covered, among others, the transfer of geolocation data, measurement of degrees of anonymity, private browsing, tracking of users via Facebook’s “Like” button, the “Do Not Track” initiative in the US and cookies in general. Also, the problem of legal and cultural differences in the perception and definition of privacy in our globalised world was approached.

It should be highlighted that the workshop resulted in an agreement to work together in a number of areas within the broader internet technical communities such the IAB, W3C, and IETF.

## 25.4 PrimeLife's Contributions to Standardisation in IETF

The Internet Engineering Task Force (IETF) is an open international community of network designers, operators, vendors, and researchers interested in the evolution of the Internet architecture. It is open to any interested individual who can register and attend IETF meetings, and can subscribe to and participate in Working Group mailing lists. The IETF Mission Statement is published in the Request for Comment (RFC) 3935<sup>24</sup> and states: “The goal of the IETF is to make the Internet work better. The mission of the IETF is to produce high quality, relevant technical and engineering documents that influence the way people design, use, and manage the Internet in such a way as to make the Internet work better. These documents include protocol standards, best current practices, and informational documents of various kinds.”

Apart from increasing discussions among IETF experts and PrimeLife partners, the project's results have been picked up in two early internet drafts:

- “Privacy Preferences for E-Mail Messages,”<sup>25</sup> i.e., the icon set “Privicons” that aims at communicating the sender's preferences for handling an e-mail to the recipients (cf. Section 15.6) and
- “Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management.”<sup>26</sup> that is based on [PH10].

It remains to be seen how these documents will evolve and whether ideas from these drafts will affect internet standardisation at a later stage.

## 25.5 Conclusion and Outlook

PrimeLife has been involved in several standardisation initiatives, and several partners will stay active in this field even after the project has ended. However, work on global standardisation is a trudge whose efforts are often underestimated. Certainly vendors have commercial interest in shaping standards according to their products' needs, and therefore they can usually invest more money and time in the standardisation work than entities that do not get economic benefits. Nevertheless, it is of the utmost importance that upcoming standards consider and respect societal values and legal principles, even if this means that the results may get more complicated if they take into account the varying perspectives of different cultures.

In the field of privacy and identity management, it is deemed necessary that researchers from academia and industry, practitioners from government and enterprises as well as representatives from data protection authorities and also non-governmental authorities are empowered and encouraged to contribute to standards

---

<sup>24</sup> <http://www.ietf.org/rfc/rfc3935.txt>

<sup>25</sup> <http://tools.ietf.org/html/draft-koenig-privicons>

<sup>26</sup> <http://tools.ietf.org/html/draft-hansen-privacy-terminology>

beginning from an early stage. This crucial challenge has to be tackled so that standards evolve that are not prone to inhibit human rights such as privacy or non-discrimination.

# Chapter 26

## Best Practice Solutions

Marit Hansen

### 26.1 Introduction

The PrimeLife project has worked in various areas of privacy and identity management. Some are mainly relevant for researchers, some for practitioners in the application field, and yet others tackle upcoming policy issues that yield recommendations for policy makers. The following sections point out specific findings and results of the PrimeLife project: Firstly, we address industry as being representative for application development and service provisioning. Secondly, we give recommendations to policy makers on the European, international or national level. Finally we show bits and pieces of PrimeLife's legacy and sketch possible ways where they may be picked up and developed further. Note that we can only present a small part of PrimeLife's outcome here – we had to select some of the most interesting best practice solutions that serve as example of how PrimeLife's results are potentially valuable for other stakeholders.

### 26.2 Recommendations to Industry

PrimeLife has elaborated various concepts and developed several tools that may be of value for industry. In the previous sections, PrimeLife's results regarding open source tools and its initiatives in the field of standardisation have already been described. Both can affect industry in multiple ways. For instance, application developers are invited to look at the available open source tools or modules to check whether they may be helpful in their own projects. Further, industry may participate in the discussion on standardisation related to privacy and identity management. It is clear that a few standards will be finalised in the near future – here it can be profitable for companies to be early adopters and align their products or services to the upcoming standards. In the case of mandatory or de-facto standards, the adoption



of standards will be essential to guarantee interoperability or compliance. Privacy-enhanced application design that is in line with the developed standards can also make it easier to be awarded a privacy seal that may serve as competitive advantage.

In the following, we list a few of PrimeLife's results and best practice solutions together with the stakeholders within industry that can make best use of them.

### ***26.2.1 Data Minimisation by Pseudonyms and Private Credentials***

*Stakeholders:* application developers, service providers, IT infrastructure developers.

Since PrimeLife's preceding project "PRIME – Privacy and Identity Management for Europe", the use of pseudonyms and private credentials to combine the needs of data minimisation and accountability has been stressed. This main theme has been proven to be valid also in PrimeLife, and there are a few other applications that support working with pseudonyms and even integrate private credentials. Still, these concepts have not yet been largely picked up, although the principle of data minimisation – disclosing and processing no more personal data than necessary – is based on the European data protection law. However, there has been some progress in the last years: In addition to the Idemix system from IBM (cf. Section 24.6.1), that became part of PrimeLife, related implementations exist that offer accountability while providing data minimisation, in particular the U-Prove system from Microsoft and the new German electronic identification card.

*Recommendations for application developers and service providers:* Try to minimise personal data, offer pseudonymous or anonymous use of services, integrate private credential systems for a combination of data minimisation and accountability.

*Recommendations for IT infrastructure developers:* Since many infrastructural components work with personal data of users, think of ways to minimise these data. This is particularly relevant when setting up identity infrastructures that involve authentication or identification of individuals.

### ***26.2.2 Improvement of Privacy Functionality in Social Media***

*Stakeholders:* developers and providers of social media.

Social media have become an extremely successful application area with enormous growth in numbers of users and exchanged data. Since the essence of social media is communication among people and exchange of data related to these people, it does not work well to remind users mantra-like of the principle of data minimisation. However, improved privacy and identity management functionality for existing social media would be more than helpful. This encompasses in particular the possibility of audience segregation [Gof59], i.e., the compartmentalisation of different

social circles in which users can present themselves differently. This is very much related to the principle of purpose binding from the European data protection law or to the approach of contextual integrity [Nis04]. Right now, most users of social media do not have different accounts at the same service (note that operating multiple accounts per person is not encouraged or even excluded in the terms and conditions of many social media services), and with the offered access control possibilities, the user cannot map the usual variety of an individual's social circles. In the PrimeLife project, the social network *Clique*<sup>1</sup> has been developed that demonstrates how audience segregation can be implemented and how this functionality can be used.

Another protection of the user's content against undesired access can be realised by encryption. Within the PrimeLife project, the Firefox extension *Scramble!*<sup>2</sup> has been developed that can be used in *Clique* or other social networks. It encrypts textual content in a way that only the desired audience in possession of the fitting cryptographic key can decrypt it and read the clear text. This form of audience segregation has the advantage that the social network provider does not have access to the clear text, either. With *Clique* alone, trust in the social network provider is essential because it is technically feasible for the provider to access all content. Note that even encryption of the content is not sufficient to protect the users from being spied on by the provider of a centralised social network who has the technical ability to monitor all log-in and log-out processes and analyse the social graph between the nodes in the social network. With the approach of decentralised social networks, the amount of knowledge per provider may be reduced, but this is not necessarily a panacea as long as there are no guarantees that these smaller providers (and their nodes in the social network) are trustworthy.

*Recommendations for developers and service providers of social media:* Include appropriate privacy functionality in your system by conceptualising and implementing usable ways of privacy and identity management, in particular offer audience segregation. Encourage users to protect themselves, e.g., by encryption of content. For all functionality, make clear the privacy implications for the user, e.g., what personal data are being disclosed to whom, how the data can be erased, whether a function bears irrevocable consequences for the user or other individuals. Follow the principle of “privacy by default”, i.e., configure the settings of the system in a way that provides most and not least privacy, e.g., prefer “opt-in” by the user concerning functionality that may infringe her privacy over “opt-out”. Since most social networks are based on targeted advertising of their users, think of alternative business models that do not require the collection and profiling of so much user data.

---

<sup>1</sup> <http://www.primelife.eu/results/opensource/64-clique>, see also Sections 2.2.3 and 24.2.1.

<sup>2</sup> <http://www.primelife.eu/results/opensource/65-scramble>, see also Sections 2.2.4 and 24.2.2.

### 26.2.3 *Better Protection of the User's Privacy on the Web*

*Stakeholders:* users, employers.

Today users are not well informed about how they are tracked on the web, and for users who disapprove of being tracked, neither websites nor standard web browsers provide sufficient protection against it. Within the PrimeLife project, the Firefox extension *Privacy Dashboard*<sup>3</sup> has been developed that logs the HTTP traffic to a local database on the user's computer and offers a variety of queries for analysing the log entries. Among others, users can see whether the website they are currently visiting uses third party content or invisible images. They can also set per site preferences, e.g., whether to block third party content, persistent cookies, flash cookies or scripting.

*Recommendations for users:* Firefox users are invited to install the extension so that they can be better aware of website tracking and enforce their preferences. Users of other browsers should use similar software that protects their privacy.

*Recommendations for employers:* Employers in companies or administration should make sure that their employees are well protected against unwanted data disclosure on the Internet: Tracking their employees may not only harm their privacy, but also may disclose information regarding their work and their organisation that should be protected against industrial espionage. Tools such as PrimeLife's *Privacy Dashboard* can help to achieve a better protection by blocking undesired tracking attempts and raising the awareness of users.

### 26.2.4 *Better Information of Users on Privacy-Relevant Issues on the Web*

*Stakeholders:* service providers in all areas, including e-commerce, social media, search engines, comparison shopping sites.

PrimeLife's research on privacy policies has shown that usually they are not precise enough so that users can really understand the data processing done (or planned) by the Data Controller. In addition, many privacy policies contain legalese, which makes it difficult and hardly appealing for laypersons, i.e., typical users, to study the texts.

*Recommendations for service providers:* Service providers should offer more readable and more precise privacy policies. They may follow the approach of the Art. 29 Working Party to layered policies so that the first, immediately visible *short notice layer* contains the core information, in particular the identity of the Data Controller, the purposes of processing and any other information necessary to ensure a fair processing. Users who are interested in more detailed information can get more details from the extended versions of the privacy policy in Layer 2 (*con-*

---

<sup>3</sup> <http://www.primelife.eu/results/opensource/76-dashboard>, see also Section 24.4.

*densed notice*) and Layer 3 (*full notice*) [Par04]. The *short notice* idea has been fleshed out by the “*Send Data?*” *dialogue* mock-up developed in PrimeLife that shows users all necessary information on the planned data disclosure and possible options before transferring these data (cf. Section 14.3). Further, the understanding of the privacy policies or the user interfaces may be supported by informative privacy icons – PrimeLife invested some work to design and evaluate icons conveying information in various contexts (cf. Section 15). For e-mail or other communication services, specific *Privicons* (as proposed in cooperation with PrimeLife, cf. Section 15.6) can be used and integrated into software such as e-mail clients as well as e-mail archiving systems and the related organisational processes. As soon as standards on machine-readable policy languages or privacy icons evolve, service providers should implement them accordingly. Finally, service providers should better support users in exercising their Data Subject rights, i.e., their rights to access, rectify and erase their personal data and also the right to withdraw their consent. In PrimeLife, the Data Track was extended by online functions for users to exercise their rights as far as granted by the service providers (cf. Section 13.4). Facilitating this can contribute to the users’ acceptance and may also help to improve the data quality.

*Recommendations for providers of search engines and comparison shopping sites:* Both search engines and comparison shopping sites play an important role in the Web as they function as common entry points or gateways to the services the users are looking for. For this reason, providers of search engines and providers of comparison shopping sites should offer to evaluate privacy-specific search criteria, e.g., whether a given privacy policy is matched. Parts of this matching can already be realised by a P3P-enabled website. In addition, search engines or comparison shopping sites could evaluate self-statements or third party statements on the natural language privacy policy, on mechanisms such as the “*Send Data?*” *dialogue*, on support by privacy icons, or on awarded privacy seals.

## 26.3 Recommendations to Policy Makers

During the work on the PrimeLife project, it had become apparent that many Data Controllers do not meet the standard of the European data protection law – both inside and outside the European Union. There are multiple reasons for the lack of compliance with data protection law: Many system developers and Data Controllers are simply not aware of legal provisions, or they do not know how to implement them. For new technologies, it is often not even clear to experts in the field how to achieve compliance with the applicable regulation because the legal texts – in some cases made more than a decade ago – are phrased in a way that they do not match upcoming technologies, business models or people’s usage patterns.

The PrimeLife project did not aim at comprehensively analysing current gaps between law and technology from the privacy point of view. However, within the project, a few issues concerning today’s data protection law became apparent, and

in particular for the area of privacy and identity management throughout life, a few proposals have been elaborated.

This subsection will deal with recommendations for policy makers

- concerning clear guidelines for all stakeholders involved in conceptualising, designing, implementing and operating IT systems that can have privacy-relevant aspects,
- concerning incentives and sanctions to foster data processing compliant with data protection law, and
- concerning PrimeLife-related aspects where development of law should be considered.

In addition, a general recommendation for policy makers is of great importance, but not further elaborated in this text: For building an information society that does not sacrifice privacy and security, much more education and awareness raising is required – for citizens of all ages and skills [ENI08].

### ***26.3.1 Clear Guidelines for System Developers and Data Controllers***

For improving the level of privacy protection and for better compliance with data protection law, both system developers and Data Controllers need clear guidelines and an overview on best practices and best available techniques specific to the sector of application [ENI08]. This would not only make it easier to assess privacy and security issues in internal or external auditing, but could also lead to a reduction of privacy and security breaches and diminish the risk to individuals' privacy for the future.

Especially in PrimeLife's field of privacy and identity management for life, the requirements for designing concepts and technologies are not clear to system developers and Data Controllers both in industry and administration. Unlike the social and legal systems where society has gathered experiences for several hundreds or even thousands of years and could slowly evolve, the technological progress of our time makes it hard to keep pace. Who could have predicted the current effects of information technologies to our lives only a few decades ago? Planning terms in companies often do not go much beyond five years, but we need future-proof solutions for privacy and identity management that work for 80 or 100 years and cover all stages of life [Han10].

*Recommendations for policy makers:* Policy makers and supervisory authorities should make clear what they demand from Data Controllers, Data Processors and system developers concerning privacy-relevant data processing, i.e., how to interpret privacy regulation: As designing future-proof solutions for privacy and identity management is not an easy task, clear guidelines should be elaborated and published that refine today's legal and social requirements and enable system developers and Data Controllers to implement them accordingly. This encompasses guidelines

on how the full lifecycle of identity data has to be considered when designing IT systems, how to give sufficient information to Data Subjects and users about all privacy-relevant issues, or how delegation processes should be designed [Pri09]. In addition, examples for and references to best practices and best available techniques should be collected and published. The elaboration and regular update of all these guidelines and of the overview of best practices and best available techniques requires a defined process that involves all relevant stakeholders, in particular the data protection authorities. Further, the appropriate infrastructural components should be provided [HT10].

### ***26.3.2 Incentives and Sanctions***

The current situation where non-compliance with the European data protection standard is the rule rather than the exception is also caused by a lack of incentives for Data Controllers and, related, the rare occasion of impressive sanctions in case of complaints by Data Subjects or audits by data protection authorities in the European Union.

*Recommendations for policy makers:* Policy makers should consider incentives that encourage Data Controllers to comply with the data protection law and – even better – advance the state-of-the-art in privacy technology. This can be supported by trustworthy certification schemes and in-depth audits. Policy makers should revise the current framework for sanctioning privacy infringements and providing remedies for victims. Data protection authorities should be empowered to cover a significant share of Data Controllers with inspections and impose noticeable punishments in case of privacy infringements.

Moreover, the handling of complaints or of exercising the Data Subject rights should be improved, in particular concerning cross-border data flow. Providing standard forms for complaints or for exercising Data Subject rights in different languages, at least harmonised on the European level, could help in this respect. Further, the workflow for dealing with such requests both by Data Controllers and by data protection authorities could be standardised. This could also enable better support by information technologies and user interfaces of identity management systems as proposed by PrimeLife – culminating in possible online functions to lower the threshold for users to exercise their Data Subject rights.

### ***26.3.3 Development of Law***

The European Data Protection Directive was enacted in 1995 when the information society was in the early stages of development. Whereas the values concerning privacy and data protection that determine the legal text are still valid and seem to be accepted by a majority of citizens at least throughout Europe, the methods of data

processing have changed so much that legal provisions of that time are not always helpful in today's situations.

One example is the area of social media that PrimeLife dealt with. It is characterised by a huge growth of the amount of exchanged (personal) data and of participating users: Are all of them aware of what happens to personal data disclosed via social media? Are these social media really based on informed consent by the users with a reliability of expectations, and what happens if users would like to withdraw their previously given consent? And if there are difficulties: should the concept of consent not be revised, e.g., to limit irrevocable or unexpected consequences? Even if individuals who may disclose data on their acquaintances do not become Data Controllers in the sense of the data protection law, which would require them to meet manifold requirements,<sup>4</sup> how can privacy breaches concerning the personal data of the acquaintances be effectively prevented? How do we prevent or deal with the risk that very few social network providers, who managed to make their sites central entry points for all web activities of many individual users, have access to so much data on almost all areas of life of their members (and in some cases: also of non-members)? This also applies to other single-point-of-entry pages that are offered (and often voluntarily used), e.g., search engines.

*Recommendations for policy makers:* Policy makers should demand "privacy by default", e.g., better information of users and pre-sets for all IT systems that only minimal personal information is disclosed or transferred. They should rethink the concept of consent and possibly limit data processing based on consent in its scope or extent (e.g., consider expiry of consent after one year as a default). Further, policy makers should make clear that data processing based on consent of the Data Subject requires the Data Controller to outline the consequences of the consent, in particular to show what is revocable and what is irrevocable and how it is possible to revoke that consent. They should define and explicate areas where irrevocable consequences are limited or prohibited at all. Policy makers should further limit exemptions to use (potentially) personal data for other purposes and be extra cautious with sensitive data. Moreover, policy makers should seek ways to efficiently implement fair user control (e.g., exercising Data Subject rights in an easy and convenient way possible for all individuals). This could be supported by establishing an international warning system for specific risks or breaches, similar to governmental travel warnings for dangerous regions.

In addition, PrimeLife has worked on privacy and identity management throughout life. In principle, today's legal provisions should hold when it comes to the need for maintaining privacy for the full lifetime and in all stages of life, but there is room for improvement. In particular, the instrument of delegation to exercise Data Subject's rights should be recognised and explicated by law – this is necessary to cover all stages of life and achieve a fair balance between the interests of all parties involved.

---

<sup>4</sup> In many cases the so-called household exemption applies: The Data Protection Directive 95/46/EC does not impose the duties of a Data Controller on an individual who processes personal data "in the course of a purely personal or household activity" [Par09].



*Recommendations for policy makers:* For legally relevant settings, policy makers should regulate the circumstances of expressing and revoking delegation. Additionally, they should define general principles or guidelines for delegation that balance the interests, rights and duties of the parties involved in delegation, e.g., the person concerned, the delegates, the delegators, or other communication partners. Further, policy makers should provide prerequisites to enable later revision of privacy-relevant actions performed by the delegate on behalf of a person concerned. Regarding the protection of minors also in the area of privacy protection, the right of young persons to privacy including the right to exercise the Data Subject rights should be explicated in the law.

Lifelong privacy protection also means that policy makers can react to alterations and upcoming challenges in our changing world. Here *ex ante* privacy assessments of technical, regulatory, and legislative advancements could be helpful. Further, there should be better precautions against risks and ways to deal with them, e.g., by strengthening the principles of data minimisation (including a “right to oblivion”) and user control as well as contextual integrity to prevent information being taken out of the context [BPPB11]. Similarly, in addition to the traditional IT security protection control objectives, i.e., confidentiality, integrity, and availability, specific privacy protection control objectives should be considered and implemented as appropriate, namely transparency, unlinkability and intervenability [RP09], cf. Section 25.2.2.

*Recommendations for policy makers:* Policy makers should monitor changes in society, law and technologies and react appropriately, e.g., by evaluating chances and risks, adapting current processes, regulations or standards to the changed conditions, etc. On a micro-level, such changes that may be relevant for the individuals’ privacy can be mergers and acquisitions. Here, policy makers should consider better protection of Data Subjects and better information in case of mergers and acquisitions, in particular in cross-border mergers or mergers in third countries. Furthermore, upcoming proposals to incorporate contextual integrity as well as privacy-specific control objectives into law should be evaluated.

Finally, legal provisions may also hinder the employment of privacy-enhancing technologies. Think of private credentials that enable the disclosure of less personal data and can even prevent undesired linkage. However, there are several laws, especially in the governmental sector, that specify exactly which data are to be collected from the Data Subjects because this has been traditionally done that way for many years. This specified amount of data may have been reasonable in paper-based workflows, and now this experience has been transferred to the digital world. Still, it would be sufficient – and demanded by the data minimisation principle – if not the exact data fields, but only attributes were processed, e.g., not the exact birth date, but the proof that the Data Subject is at least 18 years old. In this example, the legal provisions themselves cement a not at all data minimising process design that could be improved if the appropriate infrastructure was provided.

*Recommendations for policy makers:* Policy makers should evaluate current legal provisions in the light of private credentials. In addition, they should support setting





## References Part VI

- [Adi08] Ben Adida. Helios: Web-based open-audit voting. In *USENIX Security Symposium*, pages 335–348, 2008.
- [Adi10] Ben Adida. Helios voting. <http://heliosvoting.org/>, November 2010.
- [BKW09] Filipe Beato, Markulf Kohlweiss, and Karel Wouters. Enforcing access control in social network sites. In *HotPETs*, 2009.
- [BPPB11] Katrin Borcea-Pfitzmann, Andreas Pfitzmann, and Manuela Berg. Privacy 3.0 := Data Minimization + User Control + Contextual Integrity. *it - Information Technology*, 53(1), January 2011. To be published.
- [CCM08] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a secure voting system. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 354–368, Washington, DC, USA, 2008. IEEE Computer Society.
- [CDCdVF<sup>+</sup>10] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. In *ACM Transactions on Information and System Security (TISSEC)*, July 2010.
- [CJ06] Kim Cameron and Michael B. Jones. Design rationale behind the identity meta-system architecture, 2006.
- [CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.
- [egr10] EGroupware – online groupware. <http://www.egroupware.org>, November 2010.
- [ENI08] ENISA. Technology-induced Challenges in Privacy and Data Protection in Europe. A report by the ENISA Ad Hoc Working Group on Privacy and Technology, European Network and Information Security Agency (ENISA), Heraklion, Crete, Greece, October 2008.
- [FS10] Tom Frey and Dominik Skrobala. moreganize – stay organised. <http://moreganize.ch>, November 2010.
- [Gof59] Erving Goffman. *The presentation of self in everyday life*. Doubleday, 1959.
- [Han10] Marit Hansen. Towards future-proof privacy-respecting identity management systems. In Norbert Pohlmann, Helmut Reimer, and Wolfgang Schneider, editors, *ISSE 2010 – Securing Electronic Business Processes*, pages 182–190. Vieweg + Teubner Verlag, 2010.
- [HT10] Marit Hansen and Sven Thomsen. Lebenslanger Datenschutz – Anforderungen an vertrauenswürdige Infrastrukturen. *Datenschutz und Datensicherheit (DuD)*, 34(5):283–288, 2010.

- [KB09] Benjamin Kellermann and Rainer Böhme. Privacy-enhanced event scheduling. In *CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering*, pages 52–59. IEEE Computer Society, 2009.
- [Kel11] Benjamin Kellermann. Privacy-enhanced web-based event scheduling with majority agreement. In Jan Camenisch, Simone Fischer-Hübner, and Yuko Murayama, editors, *SEC 2011 – Future Challenges in Security and Privacy for Academia and Industry*, IFIP Advances in Information and Communication Technology. Springer, 2011.
- [LXH09] Wanying Luo, Qi Xie, and Urs Hengartner. FaceCloak: An architecture for user privacy on social networking sites. In *CSE (3)*, pages 26–33. IEEE Computer Society, 2009.
- [Mof10] Jon Moffet. PHP Detector library. <http://phpcode.mypapit.net/demo/detector/detector.zip>, November 2010.
- [Näf10] Michael Näf. Doodle: easy scheduling. <http://www.doodle.com>, November 2010.
- [Nis04] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(119):119 – 159, 2004.
- [ope10] OpenGroupware.org. <http://www.opengroupware.org>, November 2010.
- [Par04] Art. 29 Working Party. Opinion 10/2004 on more harmonised information provisions. 11987/04/EN, WP 100, [http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2004/wp100\\_en.pdf](http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2004/wp100_en.pdf), November 2004.
- [Par09] Art. 29 Working Party. Opinion 5/2009 on online social networking. 01189/09/EN, WP 163, [http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2009/wp163\\_en.pdf](http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2009/wp163_en.pdf), June 2009.
- [PBP10] Stefanie Pötzsch and Katrin Borcea-Pfitzmann. Privacy-respecting access control in collaborative workspaces. In M. Bezzi et al., editor, *Privacy and Identity, IFIP AICT 320*, pages 102–111, Nice, France, 2010. Springer.
- [Pen08] Boris Penck. hookup.at – meet your friends. <http://hookup.at>, March 2008.
- [Pfi01] Andreas Pfitzmann. Multilateral security: Enabling technologies and their evaluation. In Reinhard Wilhelm, editor, *Informatics - 10 Years Back, 10 Years Ahead*, number LNCS 2000, pages 50–62. Springer, 2001.
- [PH10] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management v. 0.34. <https://dud.inf.tu-dresden.de/Anon.Terminology.shtml>, August 2010.
- [Pri09] PrimeLife WP1.3. Requirements and concepts for identity management throughout life. In Katalin Storf, Marit Hansen, and Maren Raguse, editors, *PrimeLife Heartbeat H1.3.5*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, November 2009.
- [Pri10] PrimeLife WP1.2. Privacy-enabled communities demonstrator. In Stefanie Pötzsch, editor, *PrimeLife Deliverable D1.2.2*. PrimeLife, <http://www.{PrimeLife}.eu/results/documents>, February 2010.
- [Pro10] Professional Software Engineering Limited, Harwell Innovation Centre. agreeA-date: free online meeting scheduler. <http://www.agreedate.com>, November 2010.
- [RB11] Martin Rost and Kirsten Bock. Privacy By Design und die Neuen Schutzziele – Grundsätze, Ziele und Anforderungen. *Datenschutz und Datensicherheit (DuD)*, 35(1):30–35, January 2011.
- [RP09] Martin Rost and Andreas Pfitzmann. Datenschutz-Schutzziele - revisited. *Datenschutz und Datensicherheit (DuD)*, 33:353–358, 2009.
- [Sol10] Andreas Åkre Solberg. Terminplaner. <https://terminplaner.dfn.de>, November 2010.
- [Tsa10] Jonathan Tsai. plugin:doodle [dokuwiki]. <http://www.dokuwiki.org/plugin:doodle>, November 2010.

## Chapter 27

# PrimeLife's Legacy

Jan Camenisch and Marit Hansen

A project's legacy consists of three main parts: the product legacy, the process legacy, and the people legacy [CHM03]. Most parts of this book deal with the *product legacy*, i.e., the tangible outcome of PrimeLife in the form of prototypes, demonstrators, the code base, research papers, contributions to standardisation initiatives, heartbeats and deliverables. In addition, project flyers, presentations, a large body of scientific publications, PrimeLife's website, and other ways of managing the project's knowledge belong to the product legacy.

The *process legacy* encompasses the process knowledge of how the project's objectives were achieved and how the results were elaborated. This includes improved capabilities for successfully and efficiently conducting or participating in future projects. All partners in the PrimeLife consortium have set up the appropriate workflows in their organisations to work jointly together and produce results that contribute to the common vision. They have learned how to present and explain PrimeLife's results to different kinds of audiences, to exchange valuable information at conferences, at exhibitions such as the European ICT Events or the international CeBIT fair, and at meetings with other projects or cluster events.

While process legacy can be put down in writing and can be implemented in workflows, *people legacy* addresses the gained contacts of individual team members as a result of successful networking as well as tacit expert knowledge gathered throughout the project's lifetime. Among others, this knowledge has been built up because all members of the PrimeLife team have practised throughout the project how to work together among several disciplines, how to find a common language, how to bridge cultural differences, and how to deliver results that meet each other's expectations. This is not only true for the people involved in each partner organisation for the full duration or parts of it, but also for the members of the PrimeLife Reference Group, for attendants of the two PrimeLife Summer Schools or for further participants of the workshops PrimeLife has organised. The networking effects of such projects should not be underestimated, because they support further evolution of PrimeLife's stimuli that may come into blossom only a few years later.

One of the spin-off projects of PrimeLife has already started: “ABC4Trust – Attribute-based Credentials for Trust”<sup>1</sup> will work from 2010 to 2014 on using data minimising certificates such as IBM’s identity mixer (idemix) credentials and Microsoft’s U-Prove system in identity pilots. Also, we already know that results from the PrimeLife project will find their ways in some of the forthcoming Future Internet Public Private Partnership projects funded by the EC. Other communities formed within the project will continue to exist, e.g., the core team of the Summer School will try to organise similar events after the project has ended. Finally, the individual participants of the PrimeLife project will be ready to answer questions concerning their specific field of research and participate in further discussions and new smaller and larger projects to promote privacy and identity management as PrimeLife understands it.

## References

- [CHM03] Lynne P. Cooper, Michael H. Hecht, and Ann Majchrzak, *Managing a project’s legacy: implications for organizations and project management*, IEEE Aerospace Conference 2003, Pasadena, CA : Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2003, <http://hdl.handle.net/2014/39362>.

---

<sup>1</sup> <http://www.abc4trust.eu/>

# Index

- $\ell$ -diversity, 160
- $\mu$ SD, 418, 425
- $k$ -anonymity, 160
- $k$ -grouping, 177
- $k$ -looseness, 177
- $t$ -closeness, 161
- identity management
  - data asset, 447, 451
- access control, 37, 46, 485, 486
  - attributes, 486
  - credential-based, *see* credential-based
    - access control
  - forums, 50, 53, 464
  - obligation, *see* obligation
  - policy, *see* policy languages
  - pseudonymous, 300
  - selective, 47, 48, 50
  - sticky policy, *see* policy languages, sticky
- access rights, 229
  - delegation of, 100
- anonymity, 20
- anonymity revocation, 120, 125
  - oblivious, 132
- anonymous
  - communication, 126
    - access, 138, 139, 142
    - communication, 66, 104, 477
    - credential, 104, *see* credential, 472
  - anonymous credential, *see* credential,
    - anonymous
- area of life, 88
- attribute, 15, 119
  - based credential selection, 240
  - certified, 13, 119
  - proving properties of, 120
  - selective disclosure, 235
  - visibility, 176
- audience segregation, 36, 39, 40, 42
  - access control, 37
- authorisations, 321, 365
- backup prototype, 99–110, 227, 228
- base encryption layer, 195, 196
- BEL, *see* base encryption layer
- best practice solutions, 492
- biometrics, 16
- blind signatures, 123
- blog, 63
  - trustworthy, 76
- bound URL, 78
- card-based credential selection, 238
- certificate, *see* credential
- Clique, 42, 460, 495
  - audience segregation, 42
  - collections, 42–44
  - faces, 42, 45
  - fine grained options, 42
  - Scramble! use, 48
  - visibility setting, 45
- collaborative design, 232
- collaborative workspaces, 35, 37
- collections, 42–44
  - Clique, *see* Clique, collections
  - Scramble!, *see* Scramble!, collections
- confidentiality, 8
- confidentiality constraint, 165
- context, 9, 13
- context collision, 39, 40, 43
- context management, 13
- contextual integrity, 20, 50, 344, 495, 501
- credential, 108

- anonymous, 13, 53, 119, 225, 227, 233, 234, 242, 307, 310, 327, 472, 494
- encryption of attributes, 120
- hardware binding, 122
- ontologies, 331
- private, *see* credential, anonymous
- revocation of, 121
- selection
  - attribute-based, 240
  - card-based, 238
  - store, 356, 360
  - use limitation, 122
- credential-based access control, 308, 328, 336, 359, 369, 473, 488
- critical incidents, 231
- data asset, *see* identity management, data asset
- data consumer, 183
- data controller, 138, 139, 313, 321, 496, 498
- data handling, 224
- data handling policy, *see* policy languages
- data minimisation, 8, 20, 233, 235, 236, 298, 300, 484, 494, 501
- data outsourcing, 181
  - access control, 183
  - authorisation policy, 183
- data owner, 183
- data processing, 298
- data protection
  - authorities, 499
  - directive, 136, 345, 499
  - legislation, 343
- Data Protection Working Party, 263
- data storage, 224
- data subject, 9, 138, 139, 304, 313, 321, 356
  - rights, 136, 484, 499
- Data To Transfer sidebar, 264
- Data Track, 253, 256–258, 466, 497
- data usage control, 489
- delegation, 91, 94, 100, 106–109, 332, 501
- digital footprint, 91
- digital traces, 227
- direct anonymous attestation, 123
- directive
  - data protection, *see* data protection directive
  - e-Privacy, *see* e-Privacy directive
  - privacy and electronic communications, 345
- disclosure, 160
  - attribute, 160
  - identity, 160
- display privacy aspects, 279
- downstream
  - data controller, 269, 313
  - matching, 321
  - usage, 313, 321, 357, 365, 384
  - usage requirements, 314
- DPD, *see* data protection directive
- DPEC, *see* directive, privacy and electronic communications
- Dudle, 464
- dynamic accumulators, 121
- e-cash, 124
- e-Privacy directive, 136
- eCV, 108, 404, 429
- Elgg, 42, 460
- enabler
  - economic valuation of identity management, 432
  - identity management, 432, 445
- encryption, 167
  - homomorphic, 126
  - searchable, 126
  - two-layered, *see* over-encryption
  - verifiable, 120
- encryption policy, 184, 187
  - correct, 188
  - graph, 187
  - key agreement function, 185
  - key assignment function, 186
  - key derivation function, 185
- EPAL, 344
- event scheduling, 464
- eXtensible Access Control Markup Language, *see* XACML
- Facebook, 35, 37–39, 41, 43, 46, 48, 352, 460
- faces, 42, 45
- forum, 36, 50, 63
- forums, 35, 37
- fragmentation, 167
  - correctness, 168, 172
  - maximal visibility, 168
  - minimal, 169, 173
- functional capability, *see* identity management, functional capability
- functional differentiation, 344
- geolocation, 488–490
- group signatures, 125
- guidelines
  - data controllers, 498
  - system developers, 498
  - usability, *see* usability guidelines
- HCI Challenge, 222–228
  - complex mechanisms, 225, 228–230
  - limited user knowledge, 222

- privacy secondary, 224
  - technology driven development, 223
  - understanding terms, 223
  - wrong mental model, 223
- heterogeneity, 178
  - association, 178
  - deep, 178
  - group, 178
- heuristic evaluation, 230
- household exemption, 500
- icon, 226, 228, 279, 344, 353, 497
  - icon set for general usage, 281
  - icon set for social networks, 282
  - PrimeLife icon set, 281
  - sets, 279
  - test, 282
- icon set, *see* icon
- idemix, *see* identity mixer, 494
- identifier, 159
- identity, 10
  - attribute, 91
  - core, 17
  - partial, 11, 13, 105, 414, 416, 429
- identity management, 10, 227
  - data asset, 432, 446
  - functional capability, 432, 446, 447, 451
  - interoperability with reputation, 153
  - privacy-enhancing, 9
- identity mixer, 53, 234, 472, 475
- IEC, 480
- IETF, 480, 491
- incentive
  - for privacy, 499
  - system, 66, 69, 78
- information security management system, 482
- informational self-determination, 484
- interface design, *see* user interface design
- International Electrotechnical Commission, *see* IEC
- International Organization for Standardization, *see* ISO
- Internet Engineering Task Force, *see* IETF
- intervenability, 484, 501
- ISMS, *see* information security management system
- ISO, 480
  - ISO 24760, 480, 481
  - ISO 29100, 480
  - ISO 29101, 480, 482
- key and token graph, 186
- lazy matching, 324
- legacy, 505
  - people, 505
  - process, 505
  - product, 505
- legislation, 343
- Liberty, 344
- limited user knowledge, 222, 225, 229, 230
- linkability, 92, 227
- log, *see* privacy preserving secure log
- loose association, 176
- machine-readable privacy statements, 285
- management
  - identity, *see* identity management
  - privacy preference, *see* privacy preference management
  - privacy, on the fly, 266
- MediaWiki, 80
- mental model, 223, 224, 226, 227, 229, 231, 232, 234
- metadata, 68
  - binding to data, 68, 71
- metric
  - privacy, 159
  - risk, 159
  - trust, 306
- mix networks, 126
- mobile device, 413
- mobile services, 417
- non-verbal cues, 231
- obligation, 108, 315, 317, 348, 359, 364, 371, 488
- oblivious transfer, 127
  - priced, 128, 129
  - with access control, 128
- oblivious trusted third party, 130
- on the fly privacy management, 262, 266
- online shopping, 350
- ontology, 347, 348, 365
- open source, 474
- OpenPGP, 47
- OpenID, 328
- OpenPGP, 191
- Over-Encrypt, 196, 471
- over-encryption, 195, 196
  - collusion attack, 196
- P3P, 299, 309, 344, 347, 354
- partial identity, 11, 13, 414, 416, 429
  - requirements for, 12
- partonomy, 343, 347, 348
- peer surveillance, 40



- perceived privacy index, 75
- performance measures, 231
- persistence, 359
- personal data, 50
- personal data MOD, 55, 462
- PET related terms, *see* privacy, terms
- PET-USES, 213, 275
- petitions, 127
- phpBB, 51, 462
- PII selection, 399
- platform for privacy preferences, *see* P3P
- PLING, 485
- policy
  - encryption, *see* encryption policy
  - language, *see* policy languages
  - legal, 343
  - management, 265
- policy languages, 261, 292, 295, 399
  - access control, 53, 296, 303, 308, 314, 327
  - administration, 261, 265
  - composition, 307
  - credential-based access control, 308, 328, 336, 359, 361, 369
  - data handling, 295, 300, 305, 309, 314, 346, 362
  - dialog management, 333
  - editor, 357
  - engine, 355
  - management, 265
  - matching, 271, 315, 357, 373, 399
  - obligation, *see* obligation
  - preferences, 314, 355, 363
  - PrimeLife Policy Language, 108, 326, 355, 360
  - privacy-preserving access control, 327, 336
  - requirements, 295, 299, 300, 303, 305
  - sanitisation, *see* policy languages, dialog management
  - sticky, 299, 314, 346, 363, 372, 487
  - store, 359
  - trust, 296, 305
  - types, 295
  - usage control, 315
  - user interface, 262
- Policy Languages Interest Group, *see* PLING
- polling, 127
- PPL, *see* policy, PrimeLife Policy Language
- PPX, 75
- preference, policy, *see* policy languages, preferences
- preference, privacy, *see* privacy, preference
- Pri-Views, 470
- PRIME, 6, 7, 9, 51, 80, 235, 245, 345, 350, 451, 488
- privacy, 8
  - unknown audience, 36
  - awareness, 34, 46, 51, 55, 462
  - best practice solutions, 492
  - breach, 305
  - by default, 495, 500
  - by design, 231
  - copying personal information, 34
  - dashboard, 229
  - development of law, 499
  - display aspects, 279
  - icon, *see* icon
  - in databases, 470
  - label, 226
  - lifelong, 17, 87, 501
  - log, *see* privacy preserving secure log
  - machine-readable statements, 285
  - management, 224
  - metric, 159
  - patterns, 226, 227
  - policy, 223, 227, 261, 279
  - preferences
    - selection, 266
  - preference, 223, 227, 314, 491
  - preferences, platform for, *see* P3P
  - protection, 224, 227
  - secondary, 224, 227, 230
  - service-oriented architecture, *see* SA383
  - settings, 39, 226
  - social networks, 34, 38–41
  - storage, 34, 40
  - terms, 223, 226, 229, 230
  - understanding terms, 226
  - unknown audience, 34, 35, 38
  - workshop on Internet privacy, 490
- Privacy Dashboard, 466, 496
- privacy management
  - on the fly, 262, 266
- privacy preference management
  - Bookmarks-based approach, 266
  - HCI, 261
  - SOA, 399
- privacy preserving secure log, 135, 138, 139
  - API, 142
  - attacker model, 139
  - entry structure, 141
  - properties, 138
  - secure log, 138
  - state, 140
  - storage, 141
- privacy protection goals, *see* protection goals, privacy
- Privacy Tuner, 266
- privacy-relevant data processing, 483

- private credentials, *see* credential, anonymous
- private world, 418, 422
- Privicons, 284, 353, 491, 497
- proactive matching, 323
- processing, 343
- protection goals
  - privacy, 482, 483, 501
  - security, 482, 501
- pseudonym, 13, 70, 80, 494
  - relationship, 15
  - role, 15
  - transaction, 15
- public world, 422
- public-key encryption, 225
- purpose, 298, 351, 365
- purpose binding, 484
  
- quasi-identifier, 160
- query evaluation, 170, 174
  
- rating, 66
- reader feedback, 81
- recommendations
  - for employers, 496
  - for policy makers, 497–501
  - for service providers, 496
  - for users, 496
  - to industry, 493
- redactable signatures, 126
- reputation, 64, 65, 74, 80, 81
  - author, 66, 80
  - control mechanism, 147
  - evaluation function, 147
  - function, 147
  - interoperability
    - applications, 150
    - identity management, 153
    - trust management, 152
  - learning mechanism, 147
  - network, 146
  - rating function, 147
  - requirements, 148
  - trust game, 146
  - user, 66, 74
- requirements
  - downstream usage, 314
  - legal, 297
  - policy languages, *see* policy languages, requirements
  - social networks, 36
  - visibility, 175
- revocation of credentials, 121
- right to oblivion, 501
- rights, 315
  
- ring signatures, 125
- risk metric, 159
  
- SAML, 328, 340, 474
- sanction, 499
- sanitisable signatures, 126
- Scramble!, 46, 461, 495
  - access control, 47, 48
  - and Web2.0, 50
  - collections, 48
  - key distribution, 47
  - keys, 50
  - OpenPGP, 47
- secure elements, 417, 420, 422
- secure log, *see* privacy preserving secure log
- SEL, *see* surface encryption layer
- selective access control, *see* access control, selective
- selective disclosure, *see* attribute, selective disclosure
- selective encryption, 182
- self estimation scale, 213
- self-determination
  - informational, 484
- Send Data dialogue, 248, 251, 265, 474, 497
  - HCI requirements, 268
  - Legal requirements, 268
  - Usability testing, 273
- sensitive
  - association, 166
  - attribute, 160
  - value distribution, 164
- service provider, 183, 496
  - honest-but-curious, 183
- service-oriented architecture, *see* SOA
- shopping, 350
- signature
  - blind, 123
  - group, 125
  - redactable, 126
  - ring, 125
  - sanitisable, 126
- SIM card, 420
- smart card web server, 420
- SOA
  - abstract framework, 402
  - abstract privacy framework, 395
  - client-server, 392
  - meta-data, 401
  - policy composition, 404
  - principle, 383
  - privacy, 394
  - privacy preference management, 399
  - privacy requirements, 385, 402

- service discovery, 398
- sticky policy, 400
- social
  - media, 494
  - software, 33, 34, 52, 460
- social network sites, *see* social networks
- social networks, 34, 37, 40, 50, 350, 352, 460, 477, 495
  - access control, 46
  - audience, 38
  - Clique, *see* Clique
  - context collision, 39, 40, 43
  - control over information, 41
  - Elgg, *see* Elgg
  - Facebook, *see* Facebook
  - forums, *see* forums
  - peer surveillance, 40
  - privacy, 34
  - privacy issues, 38, 40, 41
  - privacy-awareness, 46
  - recommendations, 494
  - requirements, 36
  - Scramble!, *see* Scramble!
  - self-presentation, 35, 41, 46
  - storage, 40
- Social Trust Factors, 246
- social web, 33
- stage of life, 90
- standardisation, 479, 493
  - IETF, 491
  - ISO/IEC, 480
  - W3C, 485
- sticker, 418, 421
  - active, 421
  - passive, 421, 425
- sticky policy, *see* policy languages
- surface encryption layer, 195, 196
  - Delta-SEL, 195
  - Full-SEL, 195
- Taschengeldparagraph, 350
- taxonomy, 343, 347, 348
- technologically driven PET development, 223, 226, 229, 230
- TET, *see* transparency enhancing tool
- Tor, 139
- transparency, 135, 298, 299, 306, 343, 483, 501
- transparency enhancing tool, 135, 136
- trust
  - content, 297
  - data handling, 297
  - establishment, 305
  - game, *see* reputation, trust game
  - languages, *see* policy languages
  - mechanisms, 305
  - metric, 306
  - requirements, 305
- Trust Evaluation Function, 247
- trusted execution environment, 418, 422
- trusted third party, 307
  - oblivious, 130
- trustworthy
  - blog, 76
  - content, 62, 73
- two-layerd encryption, *see* over-encryption
- UCD, *see* user-centered design
- UICC, *see* universal integrated circuit card
- universal integrated circuit card, 420
- unlinkability, 138, 139, 142, 484, 501
- usability guidelines, 229
- usage control, *see* policy languages
  - downstream, *see* downstream usage
- user
  - awareness, 222
  - control, 20, 501
  - evaluation, 230
  - reputation, *see* reputation, user
  - requirements, 229
  - research, 229
- user interface
  - design, 229
  - for PPL, 263
- user-centered design, 221, 227, 230, 231
- user-generated content, 50
- utility of data, 175
- Verfahrensverzeichnis, 348
- visibility requirement, 175
- visual concepts, 228
- voting, 127
- W3C, 480, 485
- Web 2.0, 33, 61
  - characteristics, 33
  - coll. workspaces, *see* collaborative workspaces
  - forums, *see* forums
  - Scramble!, *see* Scramble!
  - social web, *see* social networks
  - software as a service, 34
- wiki, 61, 64, 80
- World Wide Web Consortium, *see* W3C
- WSDL, 397
- X.509, 328
- XACML, 308, 316, 333, 336, 340, 344–346, 355, 360, 474, 485, 488